Dynamic Mode Decomposition with Memory

Ryoji Anzaki[*], Kei Sano[†], Takuro Tsutsui, Masato Kazui, Takahito Matsuzawa

AI Development Department, System Development Center, Tokyo Electron Ltd., Sapporo City 060-0003, Japan

Version 2

December 16, 2022

Abstract

In many realms of science and engineering, the time-evolution of a system plays a key role in analyzing, controlling and predicting the behavior of the system. Ranging from the climate data collected by the observation satellites to the classical mechanical motion of manufacturing equipment, there are vast amount of accumulated time-series data in ready-to-use formats.

One established and still developing method for time-series data analysis is the dynamic mode decomposition (DMD), a linear operation-based, model-free method proposed in 2008 by Schmid. In the DMD, especially in the context of the Koopman theory, one assumes the time-evolution model in which the variables evolve in time according to a homogeneous, constant-coefficient first-order ordinary differential equation (ODE). Although the DMD proves itself as one of the successful methods for time-series data analysis in its relatively short history, the abovementioned limitations on the time-evolution model still restrict the DMD from application to wider range of phenomena. Among such phenomena, we pay attention to the memory effects in time-series data. A system is said to have memory if its time-evolution is determined by the past states of the system. The DMD is based on the ODEs and thus cannot incorporate the memory effects.

In this paper, we present a novel method, DMD with memory (DMDm), to overcome the *memoryless* restriction on the time-evolution model in the existing DMD methods. We introduce a class of initial value problems whose solutions correspond to the eigenmode of a linear transformation. Using such a linear transformation in time domain, instead of the time-difference operator, we enlarge the DMD into a wider class of time-evolution models.

As a solid example of this idea, we utilize the Caputo fractional differential [3, 5, 22] to extend the DMD so that one can analyze the time-series data with power-low memory effects, which is seen in various phenomena e.g., viscoelastic matter, fluid dynamics with surface effects, and the mechanical slider with grease. We thus developed *fractional* DMD, a DMD-based method with arbitrary (real-value) order differential operations. We want to emphasize the fact that you can actually *optimize* the order of the equation with a standard optimization algorithm, so that one get the order of the equation that best explains the nature of the time-series data. This is also true for any DMDm method with a parameterized time-evolution model. We demonstrated the results against synthetic data, and successfully estimated the model parameters.

The proposed method is expected to be useful not only for the scientific use, but also for model estimation, control and failure detection of mechanical, thermal and fluid systems in factory machines, such as in modern semiconductor manufacturing equipment.

Keywords: dynamic mode decomposition, fractional-order derivative, time-series data

[*] Electronic Address: ryoji.anzaki@tel.com

[†] Electronic Address: kei.sano@tel.com

1 Introduction

Dynamic mode decomposition (DMD) [17, 24, 2, 18] is a model-free, linear algebra-based method for time-series data analysis. First developed by Schmid for analysis of experimental data in fluid dynamics [17], the DMD becomes popular among wide variety of scientific realms, ranging from climatology [11, 16], plasma physics [15] to dissipative quantum systems [13], as well as applications in fluid dynamics [8]. Besides its success in data-driven science, the mathematical structure of the DMD is also attracting wide attention, especially in terms of its connection with the Koopman theory [7, 12, 2].

1.1 Algorithm of DMD

We show the algorithm of the DMD in its simplest form, the *exact* DMD [24, 18]. Suppose we have a set of time points $\mathsf{T} = \{t_0, t_1, ..., t_{m-1}\}$ with $t_j > t_i$ for $j > i$. Suppose we have a collection of data, such as instantaneous observations of a system, $\{\boldsymbol{x}_0, ..., \boldsymbol{x}_{m-1}\}$. Each *observation* $\boldsymbol{x}_k \in \mathbf{R}^n (k = 0, ..., m-1)$ is a state (column) vector at time $t_k \in \mathsf{T}$. The exact DMD is formulated as the least-squares method for $\dot{\boldsymbol{x}}_k$ with the time-evolution model in continuous time,

$$\dot{\boldsymbol{x}}(t) = \mathcal{A}\boldsymbol{x}(t), \tag{1}$$

with $\mathcal{A} \in \mathbf{R}^{n\times n}$ being the parameter of continuous-time time-evolution model. For a uniformly discretized T with time interval $\Delta t > 0$, the optimization of $\mathcal{A}$ is performed by a matrix manipulation by letting $X = \left[\boldsymbol{x}_0, \boldsymbol{x}_1, ..., \boldsymbol{x}_{m-2}\right]$ and $X' = \left[\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_{m-1}\right]$, as

$$\mathcal{A} \simeq \frac{A - \hat{1}_n}{\Delta t}, \quad A = \arg\min_{A'} \|X' - A'X\| = X'X^+, \tag{2}$$

where $\hat{1}_n$ is the $n \times n$ unit matrix, and $\|\bullet\|$ is the matrix Frobenius norm. Note that, for time-series data that obey the first-order difference equation, the matrix A becomes the transition matrix: $A\boldsymbol{x}_k = \boldsymbol{x}_{k+1}$. The Moore-Penrose pseudo-inverse $\bullet^+$ can be calculated by the singular-value decomposition (SVD) $X = U\Sigma V^*$. Here $U \in \mathbf{C}^{n\times n}$ and $V \in \mathbf{C}^{m\times m}$ are unitary, i.e., $U^*U = \hat{1}_n$ and $V^*V = \hat{1}_m$, where $\bullet^*$ denotes the adjoint matrix. The matrix $\Sigma \in \mathbf{R}^{n\times m}$ contains the singular values of X. The pseudo-inverse X^+ is now calculated as $X^+ = V\Sigma^+U^*$, so that the parameter of the discrete-time time evolution model is now given by,

$$A = X'V\Sigma^+U^*. \tag{3}$$

The rank-r approximation of the SVD for a positive integer $r < n$ becomes $X \simeq U_r\Sigma_r V_r^*$ for $U_r \in \mathbf{C}^{n\times r}, \Sigma_r \in \mathbf{C}^{r\times r}$ and $V_r \in \mathbf{C}^{n\times r}$. Thus, we can get the rank-r representation of the discrete-time time-evolution model as follows:

$$A_r = U_r^*AU_r = U_r^*X'V_r\Sigma_r^+. \tag{4}$$

The low-rank representation is often used for data with large dimensions.

1.2 Time-evolution model

The time-evolution model of the existing DMD is the first-order ordinary differential equation (ODE) Eq. (1), whose generic solution is expressed as the superposition of the DMD modes (i.e., eigenvectors of the coefficient matrix $\mathcal{A}$) with time-evolution function expressed in exponential of time:

$$\boldsymbol{x}(t) = \sum_{j=0}^{n-1} \vec{\phi}_j \exp(\omega_j t) b_j = \mathbf{\Phi} \exp(\mathbf{\Omega} t)\boldsymbol{b}, \tag{5}$$

with $\mathbf{\Phi} = \left[\vec{\phi}_0, \vec{\phi}_1, ..., \vec{\phi}_{n-1}\right]$ and $\mathbf{\Omega} = \mathrm{diag}(\omega_0, \omega_1, ..., \omega_{n-1})$ being the eigenvectors and eigenvalues of $\mathcal{A}$, respectively. The vector $\vec{\phi}_j$ is the j-th DMD mode, corresponding to the DMD eigenvalue ω_j, while b_j is the loadings for each DMD mode. Note that a DMD mode and the corresponding exponential function have the same eigenvalue to satisfy the equation Eq. (1). A low-rank representation corresponds to the replacement $\sum_{j=0}^{n-1} \to \sum_{j=0}^{r-1}$ in Eq. (5) for $1 < r < n$.

Using the generic solution above, the time-evolution model Eq. (1) can be solved for a given initial value $\boldsymbol{x}_0$, via the replacement $\boldsymbol{b} = \boldsymbol{\Phi}^{+}\boldsymbol{x}_0$ in Eq. (5).

1.3 Non-uniform time points

The set of time points T can be uneven in many realistic situations. In such cases, one can also construct a discretized representation of the time-derivative operator. Hereafter, let us assume that the data matrix X is defined by $X = [\boldsymbol{x}_0, \boldsymbol{x}_1, ..., \boldsymbol{x}_{m-1}]$. The time-evolution model now becomes of the form,

$$XD_m^{(1)} = AX_{1:|0}, \tag{6}$$

with

$$X_{k:|a} = [\overbrace{\boldsymbol{a}, ..., \boldsymbol{a}}^{k}, \boldsymbol{x}_k, \boldsymbol{x}_{k+1}, ..., \boldsymbol{x}_{m-1}]; \quad \boldsymbol{a} = [a, a, ..., a]^{\top}, \tag{7}$$

and the matrix $D_m^{(1)} \in \mathbf{R}^{m\times m}$ being

$$D_m^{(1)} = \begin{bmatrix} 0 & -\theta_{01} & 0 & & 0 \\ 0 & +\theta_{01} & -\theta_{12} & & 0 \\ 0 & 0 & +\theta_{12} & \ddots & 0 \\ \vdots & & & \ddots & -\theta_{m-2,m-1} \\ 0 & 0 & 0 & & +\theta_{m-2,m-1} \end{bmatrix}, \tag{8}$$

where $\theta_{k-1,k} = 1/(t_k - t_{k-1}) > 0$. Eq. (6) shows itself as a special case of the following generic matrix equation for an integer $0 \leq q \leq m-1$ and an upper triangular matrix $D \in \mathbf{C}^{m\times m}$ such that $D_{:q} = \left[\mathbf{0}, ..., \mathbf{0}\right]$ with $\mathbf{0}$ being the zero vector:

$$XD = AX_{q:|0}, \tag{9}$$

where D is a matrix acting on the *temporal* indices of the data matrix $X \in \mathbf{C}^{n\times m}$, while $A \in \mathbf{C}^{n\times n}$ is a matrix acting on the *spatial* indices of X.

1.4 DMD with control

Among the extensions of the DMD, the DMD with control (DMDc) [10] is a method for analysis of time-series data corresponding to a non-autonomous dynamical system. The term *non-autonomous* means the existence of the exogenous external forcing, whose time-dependence is not affected by the status of the dynamical system itself. One example is the forced oscillation, in which one applies the external forcing to the oscillator. The generic form of the non-autonomous dynamical system of our interest is expressed as follows:

$$\frac{\mathrm{d}\boldsymbol{x}}{\mathrm{d}t} = A\boldsymbol{x}(t) + B\boldsymbol{u}(t), \tag{10}$$

with $\boldsymbol{x}$ being the n-dimensional state vector and $\boldsymbol{u}$ being the ℓ-dimensional external forcing vector. The coefficients $A \in \mathbf{R}^{n\times n}$ and $B \in \mathbf{R}^{n\times \ell}$ are constants in time. The DMDc is formulated as the estimation of matrices A, B from the observations X and the external forcing $\Upsilon = [\boldsymbol{u}_0, \boldsymbol{u}_1, ..., \boldsymbol{u}_{m-1}]$ as follows:

$$\left[A\ B\right] \simeq \left[\bar{A}\ \bar{B}\right] = XD_m^{(1)} \begin{bmatrix} X_{1:|0} \\ \Upsilon_{1:|0} \end{bmatrix}^{+}, \tag{11}$$

where $[A\ B]$ is a block matrix with two blocks A and B. The application of the standard procedure of singular-value decomposition (SVD) leads to the reduced representation [10].

1.5 Other Related Works

The simple, highly extensive algorithmic structure of the DMD helped and encouraged the researchers to develop better numerical methods based on the original DMD. The residual DMD (resDMD) [11] is a neural network-based method with ResNet-like blocks with its reference is replaced by the results of the DMD algorithm. The incorporation of neural nets enables resDMD to deal with time-series data that is best fitted by a highly nonlinear time-evolution models. The optimized DMD (optDMD) [1] and bagging, optimized DMD (BOP-DMD) [16] are extensions of DMD, with its nonlinear optimization gives a better de-biasing while the bagging of snapshots gives a better convergence to the optimization and enables an uncertainty quantification.

1.6 Memory Effects

Despite the huge success of the abovementioned methods, there is a room for improvements in the way the DMD incorporates the time-evolution models. The existing DMDs are either using the first-order ODE (exponential) model [24, 10, 2] or, using neural network-based method to deal with the nonlinearity in the time-evolution models [11]. As another extension of DMD, it is possible to extend the time-evolution model within the linear model but with the memory effects.

Many known fundamental physical processes are governed by the first- or second-order ODEs, leading to exponential-like behaviors in time. Thus, the exponential time-evolution plays a crucial role in various realms in physics.

Although the microscopic, fundamental physics are governed by integer-order ODEs (i.e., the memoryless equations of motion), systems with strong coupling to an external system, or a *reservoir*, behave differently. Suppose we know the internal state of the system, and we have no microscopic information on the internal state of the reservoir. Once we apply a stimulus to the system, it causes a change in the system state, leading to a change of the reservoir state via the system-reservoir interactions. Subsequently, the change in the reservoir might also affects the system back via the system-reservoir surface. This sort of *indirect* effects via an external system leads to a memory effect, in which the time-evolution of the system seems to be affected not only by the *current* status, but also by the *history* of its time-evolution [21, 3]. If the weight of the memory effect decays in time according to a power-law, it leads to an equation of motion described by fractional differential equations (e.g., [20]).

As explained above, the *memory* of a system means a nonlocal behavior in time-domain, mostly arising from limitations of our observations.

1.7 Aim of This Paper

In this paper, we propose a DMD-based method, DMD with *memory* (DMDm), whose time-evolution models are described by a wider class of equations that allows the descriptions of the system with memory effects. As the basis of our discussion, we use the definition of exact DMD given by Tu *et al.* (see Definition 1 of [24]). Let us assume a linear functional relation $y = \pi(x)$ for two time-series data $x : t \mapsto x(t)$ and $y : t \mapsto y(t)$. If the value $y(t) = \pi(x)(t)$ is *not* affected by the values $x(t')$ for $t' > t$, the linear map π is said to be *causal*, since the operation is performed without knowing the *future* values of x.

It is worth noting that the construction of an eigenfunction of a causal linear operator can also be formulated as an initial-value problem. This is also true in the discrete-time time-series data and a discretized representation of the linear operator D_π, as we will see in the next section in detail. The discretized representation of the eigenfunction $z : t \mapsto z(t)$ corresponding to an eigenvalue λ is the array of numbers $\big[z(t_0), z(t_1), ..., z(t_{m-1})\big]$, that satisfies the matrix equation:

$$\big[z(t_0), z(t_1), ..., z(t_{m-1})\big] D_\pi = \lambda \big[0, ..., 0, z(t_q), ..., z(t_{m-1})\big], \tag{12}$$

where the first q columns of the matrix D_π are zero vectors with $m - q$ being the rank of D_π. For a causal D_π, the element of the array $z(t_k)$ is constructed by an initial-value problem, or, by applying the *transition operator* $\mathcal{K}^{t_k}$ to the initial q states of the system $z(t_0), z(t_1), ..., z(t_{q-1})$. The above discussion leads to a mode decomposition of the form, similar to Eq. (5):

$$\boldsymbol{x}(t) = \sum_{j=0}^{r-1} \vec{\phi}_j F_{\pi,\lambda_j,z_{\text{ini}}}(t). \tag{13}$$

In Eq. (13), the vector $\vec{\phi}_j \in \mathbf{R}^n$ is a DMD mode of the problem, and the function $F_{\pi,\lambda,z_{\text{ini}}}$ is the solution of an initial value problem $\pi(z)(t) = \lambda z(t)$, $\big[z(t_0), z(t_1), ..., z(t_{q-1})\big] = z_{\text{ini}}$. Note that the eigenvector in spatial direction $\vec{\phi}_j$ and the eigenfunction in time domain $F_{\pi,\lambda_j,z_{\text{ini}}}$ share the common eigenvalue λ_j, corresponding to the factorization of the solution into temporal and spatial parts. Also, it is worth noting that this proposed framework contains the exact DMD, because

$$F_{\frac{\mathrm{d}}{\mathrm{d}t},\lambda,[1]} = \exp(\lambda t). \tag{14}$$

Note that, for $\pi = \frac{\mathrm{d}}{\mathrm{d}t}$ the rank of the discretized representation $D_{\frac{\mathrm{d}}{\mathrm{d}t}} \in \mathbf{R}^{m\times m}$ is $m-1$.

As a solid example of the above discussions, we use the fractional calculus [3, 21, 22, 5] for the time-domain transformation. Fractional derivative of real-valued [3, 22, 5] and complex-valued orders [14] are quite useful in physics with memory effects with power-law [21, 3]. The idea of introducing fractional (non-integer) order derivative is also attracting attention in the control theory [23, 9]. One of the notable applications of the fractional calculus for the control theory is the PI$^\lambda$D$^\mu$-controller [9].

The α-th order fractional integral of an integrable function f and a real value $\alpha > 0$ is given by the following [22, 5].

$$(I^\alpha f)(t) = \frac{1}{\Gamma(\alpha)} \int_{-\infty}^{t} \mathrm{d}t' \, \frac{f(t')}{(t-t')^{1-\alpha}}. \tag{15}$$

The fractional integral satisfies the following properties for any integrable functions f, g and real values $\alpha, \beta > 0$:

1. $I^\alpha(f+g) = I^\alpha f + I^\alpha g$,
2. $I^\alpha I^\beta f = I^{\alpha+\beta} f$,
3. $I^1 f(t) = \int_{-\infty}^{t} \mathrm{d}t' \, f(t')$.

The first and second conditions correspond to linearity with respect to the function and additivity for the order, respectively. The third condition is the equivalence of I^α to the Riemann integral for the case $\alpha = 1$. Noting that the integral is the inverse operation of the derivative, one can construct α-th order derivative for any $\alpha \in \mathbf{R}$ [22] for a smooth, integrable function $f : (-\infty, t_1] \to \mathbf{R}$ $\quad (t_1 > -\infty$ is the upper bound of the domain of $f)$.

In the following text, we will see a concrete definition for the α-th order fractional derivative $\frac{\mathrm{d}^\alpha f}{\mathrm{d}t^\alpha}$ for an integrable function $f : \mathbf{R} \supset [t_0, t_1] \to \mathbf{R}$ and show the idea of *fractional* DMD, or fracDMD, in which fractional differential equations determine the time-dependency of each DMD mode, instead of the first-order differential equations.

The remainder of this paper is organized as follows: in Section 2, we investigate the property of a causal linear functional and its eigenfunctions. In Section 3, we introduce the DMDm, and in Sections 4-6 we introduce the arbitrary order DMD (fracDMD) as an example of the DMDm. In Sections 7, 8, we apply the fracDMD to a synthetic time-series data to show the validity of the proposed method. Section 9 is devoted for the discussions of the numerical results, and in Section 10 we conclude the article.

Throughout this paper, we denote any complex matrix by $\bullet$, and any scaler by $-$. The column vector $\bullet_k$ and slice $\bullet_{k:\ell}$ of a matrix for $0 \le k \le \ell \le m$ are defined as, for a matrix $M = \big[\mathbf{m}_0, \mathbf{m}_1, ..., \mathbf{m}_{m-1}\big] \in \mathbf{C}^{n\times m}$,

$$M_k = \mathbf{m}_k, \quad M_{k:\ell} = [\mathbf{m}_k, ..., \mathbf{m}_{\ell-1}]. \tag{16}$$

Also, we use $M_{k:} = M_{k:m}$ and $M_{:\ell} = M_{0:\ell}$ for short. We denote the a-padded matrix for a given matrix $M = \left[\mathbf{m}_0, \mathbf{m}_1, ..., \mathbf{m}_{m-1}\right] \in \mathbf{C}^{n\times m}$ and a scaler $a \in \mathbf{C}$ by $M_{k:\ell|a}$, i.e.,

$$M_{k:\ell|0} = \big[\overbrace{\boldsymbol{a}, ...\boldsymbol{a}}^{k}, \mathbf{m}_k, ..., \mathbf{m}_{\ell-1}, \overbrace{\boldsymbol{a}, ..., \boldsymbol{a}}^{m-\ell}\big], \quad \boldsymbol{a} = \big[a, a, ..., a\big]^\top. \tag{17}$$

A block matrix with blocks $A_i \in \mathbf{C}^{n\times m}$ $(i = 0, 1, 2, ..., q-1)$ being stacked column-wise is denoted by $\left[A_0; A_1; ...; A_{q-1}\right]$, i.e.,

$$\left[A_0; A_1; ...; A_{q-1}\right] = \begin{bmatrix} A_0 \\ A_1 \\ ... \\ A_{q-1} \end{bmatrix} \in \mathbf{C}^{nq\times m}, \tag{18}$$

for short.

2 Time-Evolutions as Eigenvalue Problems

In this section, we investigate the conditions for the linear operator π introduced in the previous section. For a given set of time points $\mathcal{T} \subseteq \mathbf{R}$, let us denote the set of one-dimensional time-series data by $\mathbf{C}^{\mathcal{T}}$. A time-series datum $V \in \mathbf{C}^{\mathcal{T}}$ is a map from the set of time points $\mathcal{T}$ to complex numbers, i.e., $V : \mathcal{T} \to \mathbf{C}$.

2.1 Causal Linear Operator

For a bounded set of time points $\mathcal{T}$ and its subset $\mathcal{S} \subseteq \mathcal{T}$, let us define a *causal linear operator* $\pi : \mathbf{C}^{\mathcal{T}} \ni V \mapsto W \in \mathbf{C}^{\mathcal{S}}$ as a linear functional $\pi \in \mathrm{hom}(\mathbf{C}^{\mathcal{T}}, \mathbf{C}^{\mathcal{S}})$ between two time-series that satisfies the following *causality* condition. For a bounded set $\mathcal{T}$, we assume that $\min(\mathcal{T}) = 0$ and $\max(\mathcal{T}) = T$ without loss of generality.

- π is causal $\overset{\text{def}}{\Leftrightarrow}$ $\pi(V)(t) = \pi(V(t)\Theta(t'-t))(t)$ for any $t, t' \in \mathcal{S}$ satisfying $0 \le t \le t' \le T$,

Where $\Theta : \mathbf{R} \to \mathbf{C}$ is the step function: $\Theta(t) = 0$ for $t < 0$ and $\Theta(t) = 1$ for $t \ge 0$.

2.2 Eigenfunction of Causal Linear Operator

Let us consider a set of time points $\mathcal{T} = [0, T]$ and $\mathcal{S} = (0, T]$ for $T > 0$. For a given causal linear operator $\pi \in \mathrm{hom}(\mathbf{C}^{\mathcal{T}}, \mathbf{C}^{\mathcal{S}})$ and initial conditions, we can approximate the *eigenfunction* of π by an iterative method as follows. First, we introduce and fix a finite subset of $\mathcal{T}$ as follows:

$$\mathsf{T} = \{t_i \in \mathcal{T} | i = 0, 1, 2, ..., m-1,\ t_i > t_j \text{ for } i > j,\ t_0 = 0,\ t_{m-1} = T\} \subset \mathcal{T}. \tag{19}$$

Let us further introduce a (row) vector representation of a time-series V on T as $V = \left[V_0, V_1, ..., V_{m-1}\right]$ with $V_k = V(t_k) \in \mathbf{C}$ $(k = 0, 1, ..., m-1)$. The matrix representation of π is then introduced by the following equation:

$$VD_\pi = \pi(V), \tag{20}$$

where, in the left-hand side, a standard matrix multiplication is assumed. Due to the causality of the linear operator π, the matrix representation D_π is an upper-triangular matrix whose first q $(1 \le q \le$

$m-2$) column vectors are zero vectors:

$$D_\pi = \begin{bmatrix} 0 & \cdots & 0 & * & \cdots & * & * \\ & \ddots & \vdots & * & \cdots & * & * \\ & & 0 & * & \cdots & * & * \\ & & & \pi_{qq} & & * & * \\ & & & & \ddots & \vdots & \vdots \\ & & & & & \pi_{m-2,m-2} & * \\ \Large O & & & & & 0 & \pi_{m-1,m-1} \end{bmatrix}. \tag{21}$$

Hereafter, we assume that the product of the diagonal elements is nonzero, i.e., $\pi_{qq}\pi_{q+1,q+1}\cdots\pi_{m-1,m-1} \neq 0$. The integer q is called the *nullity* of the operator π. Note that the zeroth column vector of the matrix D_π corresponds to $t = 0 \notin \mathcal{S}$.

The eigenvalue problem for the linear operator π is approximately expressed as the following matrix equations together with the initial condition for the row vector V and the matrix D_π:

$$VD_\pi = \lambda V_{q:|0}, \quad V_{:q}\Omega = \boldsymbol{\gamma}, \tag{22}$$

where $\Omega \in \mathbf{C}^{q\times p}$ is a matrix and $\boldsymbol{\gamma} \in \mathbf{C}^p$ is a constant row vector. For the initial value problem Eq. (22) to be solvable, one must impose that $\mathrm{rank}(\Omega) = q$. Note that the second equation in Eq. (22) corresponds to the initial conditions, specifying the first q components of the vector V, while the first equation specifies the other components $V_q, V_{q+1}, ..., V_{m-1}$ based on the first q values. The simplest class of the initial condition is to give the first q values of the time-series V:

$$V_{:q} = [V_{\mathrm{ini},0}, V_{\mathrm{ini},1}, ..., V_{\mathrm{ini},q-1}], \tag{23}$$

corresponding to the case $\Omega = \hat{1}_q$ and $\boldsymbol{\gamma} = [V_{\mathrm{ini},0}, V_{\mathrm{ini},1}, ..., V_{\mathrm{ini},q-1}]$ in Eq. (22).

The eigenfunction of the operator π is now numerically approximated by the following successive calculations for given initial condition: for $k = 0, 1, ..., q-1$,

$$V_k = \sum_{j=0}^{p-1} \left(\Omega^+\right)_{jk} \gamma_j, \tag{24}$$

and for $k = q, q+1, ..., m-1$,

$$V_k = \left[\frac{V_{:k|0}D_\pi}{\lambda - \pi_{kk}}\right]_k, \tag{25}$$

The procedure to obtain the approximated eigenmode of a causal linear operator π from first q values $V_0, V_1, ..., V_{q-1}$ is summarized in Alg. 1.

Algorithm 1 Approximated eigenfunction $\boldsymbol{\phi}_{\pi,\lambda,V_{\mathrm{ini}}}$ for a causal linear operator π

```
Require: Nullity of the operator $q$, parameter $\lambda$, initial values $V_{\mathrm{ini}} = \left[V_0, V_1 ..., V_{q-1}\right] \in \mathbf{C}^q$, matrix representation $D_\pi$ corresponding to the discrete set of time points $\mathsf{T}$
  $V \leftarrow [V_0, V_1, ..., V_{q-1}, 0, ..., 0] \in \mathbf{C}^m$
  for $k \in \{q, q+1, ..., m-1\}$ do
    $V_k \leftarrow \left[\frac{V_{:k|0}D_\pi}{\lambda - \pi_{kk}}\right]_k$
    $V \leftarrow \left[V_0, V_1, ..., V_k, 0, ..., 0\right] \in \mathbf{C}^m$
  end for
  $\boldsymbol{\phi}_{\pi,\lambda,V_{\mathrm{ini}}} \leftarrow V$
  return $\boldsymbol{\phi}_{\pi,\lambda,V_{\mathrm{ini}}}$
```

Using a sequence of time-points T with its maximum time interval $\Delta t_{\max} \to +0$, Alg. 1 can approximate a smooth function. Note that, for Alg. 1 to work properly, we must find a matrix D_π whose first q columns are zero vectors while its other column vectors are linearly independent.

3 Dynamic Mode Decomposition with Memory

Using causal linear operator and their eigenfunctions introduced in the previous section, we now postulate the idea of the DMD with memory (DMDm). The DMDm is a new method to include the memory effects in the DMD framework. It employs a causal linear operator instead of the difference operator used in the existing DMD. This gives us the ability to handle the effects from the past data in the time-evolution model without losing advantages of the DMD framework. Throughout this section, we define intervals $\mathcal{T}, \mathcal{S} \subset \mathbf{R}$ by $\mathcal{T} = [0, T]$ and $\mathcal{S} = (0, T] \subset \mathcal{T}$ for $T > 0$.

3.1 Definition of the Model

Let us assume that a linear functional $\pi : \mathcal{T} \to \mathcal{S}$ is causal. We introduce a linear time-evolution model of the form,

$$\pi(\boldsymbol{x})(t) = A\boldsymbol{x}(t) \quad (t \in \mathcal{S}), \tag{26}$$

where $A \in \mathbf{R}^{n\times n}$ is a constant matrix. Restricting t onto the (finite) set of time points $\mathsf{T} = \{t_i \in \mathcal{T} | i = 0, 1, ..., m-1,\ t_0 = 0,\ t_{m-1} = T,\ t_i > t_j \text{ for } i > j\} \subset \mathcal{T}$, we can discretize this continuous-time time-evolution model to get the matrix representation

$$XD_\pi = AX_{q:|0}, \tag{27}$$

where we have a matrix A and n-dimensional time-series data X, instead of a scaler λ and a 1-dimensional time-series V in the first equation in Eq. (22), respectively. The nullity q is dependent on the nature of the operator π. Given the SVD $X = U\Sigma V^*$ with unitary matrices U and V, we can transform the Eq. (27) as follows:

$$U^*XD_\pi = U^*AU\big(U^*X_{q:|0}\big). \tag{28}$$

For a diagonalizable matrix $A \sim \text{diag}(\lambda_0, ..., \lambda_{n-1})$, we can always find an array of new variables $\Xi = \mathcal{U}^{-1}U^*X$ (i.e., we define a new variable by $\boldsymbol{\xi}(t) = \mathcal{U}^{-1}U^*\boldsymbol{x}(t)$ and $\Xi = \big[\boldsymbol{\xi}_0, \boldsymbol{\xi}_1, ..., \boldsymbol{\xi}_{m-1}\big]$) for an appropriate matrix $\mathcal{U}$ such that:

$$\Xi D_\pi = \text{diag}(\lambda_0, ..., \lambda_{n-1})\Xi_{q:|0}. \tag{29}$$

The above equation leads to the solution for the original variable $\boldsymbol{x}$ of the following form

$$\big[\boldsymbol{x}(t_0), \boldsymbol{x}(t_1), ..., \boldsymbol{x}(t_{m-1})\big] = U\mathcal{U}\begin{bmatrix} \boldsymbol{\phi}^\top_{\pi,\lambda_0,\xi_{0,\text{ini}}} \\ \vdots \\ \boldsymbol{\phi}^\top_{\pi,\lambda_{n-1},\xi_{n-1,\text{ini}}} \end{bmatrix} = \sum_{i=0}^{n-1} (U\mathcal{U}\mathbf{e}_i)\boldsymbol{\phi}^\top_{\pi,\lambda_i,\xi_{i,\text{ini}}}, \tag{30}$$

where the column vector $\boldsymbol{\phi}_{\pi,\lambda_0,\xi_{i,\text{ini}}}$ is defined in Alg. 1 and $\mathbf{e}_i = \big[\delta_{0i}, \delta_{1i}, ..., \delta_{m-1,i}\big]^\top$ with the unit tensor δ_{ij}, and $\xi_{i,\text{ini}} = \big[\xi_{i,0}, \xi_{i,1}, ..., \xi_{i,q-1}\big]$ is a vector consists of the values of $\xi_i = \mathbf{e}_i \cdot \boldsymbol{\xi}$ at first q time points. Note that the above expression is similar to Eq. (13).

3.2 Model Fitting via DMD Scheme

Our goal is to find an appropriate matrix A with which one can explain the data X on a discrete set of time points T. We assume the model Eq. (27) for the observed data X. The best fit parameter A that achieves a least-square for $\pi(x)(t)$ is estimated as,

$$A = \arg\min_{A'} \|XD_\pi - A'X_{q:|0}\| = XD_\pi X^+_{q:|0}. \tag{31}$$

Similar to Eq. (3), we can compute A using the SVD $X_{q:|0} = U\Sigma V^*$ to get the explicit expression for the DMD with memory:

$$A = XD_\pi V\Sigma^+U^*. \tag{32}$$

The low-rank approximation can also be performed in a similar way as in Eq. (4).

3.3 DMDc with Memory

The DMDc scheme is readily applicable for the proposed method. In the resultant method, DMD with control and memory (DMDcm), one assumes the following time-evolution model for a causal linear operator $\pi : \mathcal{T} \to \mathcal{S}$.

$$\pi(\boldsymbol{x})(t) = A\boldsymbol{x}(t) + B\boldsymbol{u}(t) \quad (t \in \mathcal{S}), \tag{33}$$

which corresponds to the matrix representation

$$XD_\pi = AX_{q:|0} + B\Upsilon_{q:|0}, \tag{34}$$

where $A \in \mathbf{C}^{n\times n}$ and $B \in \mathbf{C}^{n\times \ell}$ are coefficient matrices and q is the nullity of D_π. The DMDc prescription gives a direct calculation of the optimal coefficients, similar to Eq. (11), as

$$\begin{bmatrix} A & B \end{bmatrix} = XD_\pi \begin{bmatrix} X_{q:|0} \\ \Upsilon_{q:|0} \end{bmatrix}^+. \tag{35}$$

Alg. 1 is readily applicable for multi-dimensional case, and the time-evolution of the model Eq. (34) for a given initial condition $X\Omega = \Gamma$, external forcing Υ, and coefficient matrices A, B is obtained by the following successive calculations: for $k = 0, 1, ..., q-1$,

$$X_k = \sum_{j=0}^{p-1} \left(\Omega^+\right)_{jk} \Gamma_j, \tag{36}$$

and for $k = q, q+1, ..., m-1$:

$$X_k = \left[\frac{(AX_{:k|0} + B\Upsilon_{:k|0})D_\pi}{A - \pi_{kk}\hat{1}_n}\right]_k. \tag{37}$$

where the fraction of matrices $G \in \mathbf{C}^{n\times m}$ and $H \in \mathbf{C}^{n\times n}$ are defined by $G/H = H^{-1}G$. As is easily seen, this is a natural extension of DMDc, justifying the name DMDcm.

4 Integer-order Dynamic Mode Decomposition

We use the finite difference method for the first-order differential equation Eq. (10) in the original DMD framework. We can use a similar procedure for a-th order differential equation for $a = 2, 3, 4, ...$ as well. The resultant method is the arbitrary integer-order DMD (intDMD), as described below.

$$A = X\Delta_m^{(a)} X_{:m-\ell}^+, \tag{38}$$

with the matrix representation of the ℓ-th order derivative $\Delta_m^{(\ell)} \in \mathbf{R}^{m\times(m-\ell)}$. Although the extension to a positive-integer order seems to be nontrivial, the same effect is actually achieved by introducing auxiliary variables $X^{(n)} = X\Delta_m^{(n)}$ for $n = 1, 2, ..., a-1$ and using the extended matrix $\mathbf{X} = [X; X^{(1)}; ...; X^{(a-1)}]$ in the DMD framework.

5 Fractional Dynamic Mode Decomposition

The order of the differential operator can be extended to any real-valued numbers [22, 5]. The term for this generalized differentiation, *fractional derivative*, is somehow misleading, since we can also specify an irrational number $\alpha \in \mathbf{R}\backslash\mathbf{Q}$ as the order of the differentiation. One of the definitions for the α-th order differential of a smooth integrable function $f : \mathbf{R} \to \mathbf{R}$ is the Caputo derivative, as shown in the followings: for $t > 0$,

$$\frac{\mathrm{d}^\alpha f}{\mathrm{d}t^\alpha} = \begin{cases} \frac{1}{\Gamma(\nu)} \int_0^t \mathrm{d}t' \, \frac{f^{(n_\alpha^+)}(t')}{(t-t')^{1-\nu}} & (\alpha \notin \mathbf{Z}_{\geq 0},\ n_\alpha^+ = \max(0, \lceil\alpha\rceil),\ \nu = n_\alpha^+ - \alpha), \\ f^{(\alpha)}(t) & (\alpha \in \mathbf{Z}_{\geq 0}) \end{cases} \tag{39}$$

where $f^{(\ell)}$ is the ℓ-th order derivative of the function f for $\ell \in \mathbf{Z}_{\geq 0}$ and the ceiling $y = \lceil x \rceil$ is the minimum integer $y \in \mathbf{Z}$ such that $y - x \geq 0$. Strictly speaking, the exception for $\alpha \in \mathbf{Z}_{\geq 0}$ (the second line in Eq. (39)) is not necessary, because the two cases in Eq. (39) coincide at $\alpha \to \alpha_0 \in \mathbf{Z}_{\geq 0}$; both of them give $\frac{\mathrm{d}^\ell f}{\mathrm{d}t^\ell} = f^{(\ell)}$ for $\ell \in \mathbf{Z}_{\geq 0}$. Note that the right-hand side of the definition Eq. (39) with $\alpha \notin \mathbf{Z}_{\geq 0}$ coincides with the definition of the fractional *integral* in Eq. (15) with a replacement $\alpha \mapsto \nu$ and $f \mapsto f^{(\ell)}$ and $0 < \nu < 1$. In this section, we consider real-valued functions for simplicity.

One can construct the corresponding *eigenmode* for the α-th order differential operator with an initial condition. To see this, let us denote a discrete representation of a generic-order fractional differential operator $\frac{\mathrm{d}^\alpha}{\mathrm{d}t^\alpha}$ by $\mathcal{D}_m^{(\alpha)}$ for $\alpha \in \mathbf{R}$.

If a square matrix $D_m^{(\ell)} \in \mathbf{R}^{m\times m}$ satisfies the following equation for $N \in \mathbf{Z}_{\geq 0}$, let us call $D_m^{(\ell)}$ a N-th order approximation of ℓ-th order integer-order derivative: for a smooth function $f : \mathbf{C} \to \mathbf{C}$ and a finite set of time points $\mathsf{T} = \{t_i \in \mathbf{R} | i = 0, 1, ..., m-1, t_i > t_j \text{ for } i > j\}$,

$$D_m^{(\ell)} \begin{bmatrix} f(t_0) \\ \vdots \\ f(t_{\ell-1}) \\ f(t_\ell) \\ \vdots \\ f(t_{m-1}) \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ f^{(\ell)}(t_\ell) + \mathcal{O}\big((\Delta t_{\max})^N\big) \\ \vdots \\ f^{(\ell)}(t_{m-1}) + \mathcal{O}\big((\Delta t_{\max})^N\big) \end{bmatrix}, \tag{40}$$

where $\Delta t_{\max} = \max(\{t_{i+1} - t_i | i = 0, 1, ..., m-2\})$ is the maximum time interval and $\mathcal{O}$ is a big-O notation. Note that $\mathrm{rank}(D_m^{(\ell)}) = m - \ell$. The explicit expression of $\mathcal{D}_m^{(\alpha)}$ is now constructed using the Caputo fractional differential [22] as follows,

$$\mathcal{D}_m^{(\alpha)} = D_m^{(n_\alpha^+)} \Big[\boldsymbol{w}_0^{(n_\alpha^+ - \alpha)}, \boldsymbol{w}_1^{(n_\alpha^+ - \alpha)}, ..., \boldsymbol{w}_{m-1}^{(n_\alpha^+ - \alpha)}\Big]. \tag{41}$$

At the lowest order, the *weight* $\boldsymbol{w}_k^{(\nu)}$ is approximated as follows:

$$(\boldsymbol{w}_k^{(\nu)})_i = \begin{cases} \frac{1}{\Gamma(\nu+1)}((t_{k+1} - t_i)^\nu - (t_{k+1} - t_{i+1})^\nu) & (i \leq k) \\ 0 & (i > k) \end{cases}. \tag{42}$$

where $\Delta t_i = t_{i+1} - t_i$. The error of the approximation Eq. (42) is of $\mathcal{O}(T \Delta t_{\max})$ with $T = t_{m-1} - t_0$ being the total time. The matrix representation for the fractional derivative is now decomposed into the ℓ-th order (integer-order) differential and the $(-\nu)$-th order fractional derivative (i.e., fractional integral of order $0 < \nu < 1$). Clearly, the discretized representation $\mathcal{D}^{(\alpha)}$ satisfies the linearity and the causality, ensuring the existence of eigenmodes. In an actual numerical analysis, one might use more elaborate implementations with higher-order schemes. In the continuous limit, the eigenfunction of the fractional derivative operator is known to be expressed by the Mittag-Leffler functions [4, 6, 5].

Using the fact that $\mathrm{rank}(D_m^{(\ell)}) = m - \ell$ and $\{\boldsymbol{w}_k^{(\nu)} | k = 0, 1, ..., m-1\}$ is set of a m linearly independent vectors for $\nu \neq 1$, one can conclude that $\mathrm{rank}(\mathcal{D}^{(\alpha)}) = \mathrm{rank}(D_m^{(n_\alpha^+)}) = m - n_\alpha^+$. Thus, the discretized equation for the fractional differential equation together with an initial condition $X_{:n_\alpha^+}\Omega = \Gamma$ for $\Omega \in \mathbf{C}^{n_\alpha^+ \times p}$ and $\Gamma \in \mathbf{C}^{n \times p}$ is given by the follows:

$$X\mathcal{D}^{(\alpha)} = AX_{n_\alpha^+ :|0}; \quad X_{:n_\alpha^+}\Omega = \Gamma. \tag{43}$$

For a given time-series data $X \in \mathbf{R}^{n\times m}$ arranged in a matrix form and a fixed order of derivative α, the best fit matrix A for the model equation Eq. (43) is obtained as follows:

$$A = X\mathcal{D}^{(\alpha)} X^+_{n_\alpha^+ :|0}. \tag{44}$$

The above expression is analogous to Eq. (3) for the first-order (ordinary) DMD. The SVD of the matrix $X = \hat{U}\hat{\Sigma}\hat{V}^* \simeq \hat{U}_r\hat{\Sigma}_r\hat{V}_r^*$ with rank $0 < r \equiv \mathrm{rank}(\hat{\Sigma}_r) < n$ and unitary matrices $\hat{U}, \hat{U}_r, \hat{V}, \hat{V}_r$, diagonal matrices $\hat{\Sigma}, \hat{\Sigma}_r$ yields a low-rank representation of the dynamics for $\boldsymbol{\xi} = \hat{U}_r^*\boldsymbol{x} \in \mathbf{R}^r$,

$$\frac{\mathrm{d}^\alpha \boldsymbol{\xi}}{\mathrm{d}t^\alpha} = \Lambda\boldsymbol{\xi}, \quad \Lambda = \hat{U}_r^* X \mathcal{D}^{(\alpha)} \hat{V}_r \hat{\Sigma}_r^{-1}. \tag{45}$$

If the order α is unknown, the matrix A_* at the optimal fractional order α_* is now estimated by Alg. 2.

Algorithm 2 Grid search for order α

```
Require: Time-series data in matrix form $X$, candidates $\boldsymbol{\alpha} = \{\alpha_0, \alpha_1, ...\}$
  if $\alpha \in \boldsymbol{\alpha}$ then
    $A \leftarrow X\mathcal{D}^{(\alpha)} X_{q:|0}^+$
    $L(\alpha) \leftarrow \|X\mathcal{D}^{(\alpha)} - AX_{q:|0}\|$
  end if
  $\alpha_* \leftarrow \arg\min_\alpha (L(\alpha))$
  $A_* \leftarrow X\mathcal{D}^{(\alpha_*)} X_{q:|0}^+$
```

6 Fractional Dynamic Mode Decomposition with Control

The fracDMD proposed in the previous section is valid only for autonomous dynamical systems. In an analogy with the DMDc [10], we can extend our method to non-autonomous systems with input terms. Let us assume that $\mathcal{T} = [0, T]$ and $\mathcal{S} = (0, T] \subset \mathcal{T}$. Note that the Caputo derivative is a causal linear operator $\frac{\mathrm{d}^\alpha}{\mathrm{d}t^\alpha} : \mathcal{T} \to \mathcal{S}$. Suppose we have a dynamical system

$$\frac{\mathrm{d}^\alpha \boldsymbol{x}}{\mathrm{d}t^\alpha} = A\boldsymbol{x} + B\boldsymbol{u}(t) \quad (t \in \mathcal{S}), \tag{46}$$

with the state vector $\boldsymbol{x} \in \mathbf{R}^n$ and the external forcing (exogenous input) vector $\boldsymbol{u}(t) \in \mathbf{R}^\ell$. We construct matrices $X = [\boldsymbol{x}_0, \boldsymbol{x}_1, ..., \boldsymbol{x}_{m-1}]$ and $\Upsilon = [\boldsymbol{u}_0, \boldsymbol{u}_1, ..., \boldsymbol{u}_{m-1}]$. Similar to Eq. (11), we obtain the following estimation for the coefficients.

$$\begin{bmatrix} A \ B \end{bmatrix} \simeq \begin{bmatrix} \bar{A} \ \bar{B} \end{bmatrix} = X\mathcal{D}^{(\alpha)} \begin{bmatrix} X_{n_\alpha^+:|0} \\ \Upsilon_{n_\alpha^+:|0} \end{bmatrix}^+. \tag{47}$$

We can also make use of the SVDs, $X = \hat{U}\hat{\Sigma}\hat{V}^* \simeq \hat{U}_r\hat{\Sigma}_r\hat{V}_r^*$ and $[X; \Upsilon] = \tilde{U}\tilde{\Sigma}\tilde{V}^* \simeq \tilde{U}_p\tilde{\Sigma}_p\tilde{V}_p^*$ with $p > r$, to get an approximated low-rank representation of the dynamics for $\boldsymbol{\xi} = \hat{U}^*\boldsymbol{x} \in \mathbf{R}^r$, as follows:

$$\frac{\mathrm{d}^\alpha \boldsymbol{\xi}}{\mathrm{d}t^\alpha} = \Lambda\boldsymbol{\xi} + \Gamma\boldsymbol{u}, \quad [\Lambda \quad \Gamma] \simeq \hat{U}_r^* X\mathcal{D}^{(\alpha)} \tilde{V}_p \tilde{\Sigma}_p^{-1} \tilde{U}_p^* \begin{bmatrix} \hat{U}_r^* & 0 \\ 0 & \hat{1}_{p-r} \end{bmatrix}. \tag{48}$$

Thus, we can analyze a time-series data of input $\boldsymbol{u}(t)$ and the output $\boldsymbol{x}(t)$ for a system in the fracDMD framework, as well as for autonomous systems. The fracDMD algorithm is explicitly shown in Alg. 3. The function $\texttt{SVD}(-, r)$ is the SVD of a matrix with rank r.

Algorithm 3 FracDMD with control

```
Require: Input data in matrix form $\Upsilon$, observation data in matrix form $X$, ranks $p > r > 0$, order of
    the fractional differential equation $\alpha$
  $\hat{U}_r, \hat{\Sigma}_r, \hat{V}_r \leftarrow \texttt{SVD}(X_{n_\alpha^+:|0}, r)$
  $\tilde{U}_p, \tilde{\Sigma}_p, \tilde{V}_p \leftarrow \texttt{SVD}(\big[X_{n_\alpha^+:|0}; \Upsilon_{n_\alpha^+:|0}\big], p)$
  $X' \leftarrow X\mathcal{D}^{(\alpha)}$
  $[\Lambda \quad \Gamma] \leftarrow \hat{U}_r^* X' \tilde{V}_p \tilde{\Sigma}_p^{-1} \tilde{U}_p^* \begin{bmatrix} \hat{U}_r^* & 0 \\ 0 & \hat{1}_{p-r} \end{bmatrix}$
```

7 Quantitative Evaluations for the Models

The fracDMD, as well as the existing DMD, is to minimize the Frobenius norm of the matrix $\|X' - AX - B\Upsilon\|$, with X' being the fractional derivative $X\mathcal{D}^{(\alpha)}$ for the fracDMD. The Frobenius norm-based reconstruction error $\mathcal{L}^{(\alpha)}_{\text{Frobenius}}(A, B; X) = \|X\mathcal{D}^{(\alpha)} - AX - B\Upsilon\|$ is considered to be the sum of squared reconstruction error in α-th order time derivative, with $AX + B\Upsilon$ regarded as the reconstruction by the model.

Another way to evaluate the model is to use the sum of squared errors (SSE) in the reconstructed states $\boldsymbol{x}_i$ $(i = 0, 1, ..., m-1)$. The explicit expression of SSE $\mathcal{L}^{(\alpha)}_{\text{SSE}}(A, B; X)$ as a function of α, A, B, and the observation data $X = [\boldsymbol{x}_0, \boldsymbol{x}_1, ..., \boldsymbol{x}_{m-1}]$ is given by,

$$\mathcal{L}^{(\alpha)}_{\text{SSE}}(A, B; X) = \sum_{i=0}^{m-1} |\boldsymbol{x}(t_i) - \boldsymbol{x}_i|^2, \tag{49}$$

with $\boldsymbol{x}(t_i)$ being the solution of the time-evolution equation Eq. (46) with the coefficients A, B and the initial condition, whose explicit expression is shown in Eq. (30).

8 Numerical Experiments

We perform numerical experiments to show the utility of the fracDMD against a synthetic data. We numerically generate the solution of the fractional oscillator [19] analyzed by Svenkeson *et al.* [21] in a context of spectral decomposition. Svenkeson *et al.* performed numerical tests on the real-time behavior of a single, noise-free fractional oscillator with known parameters to show that the fractional-order calculus is useful in analyzing the memory effects. We extend their method to include multi-dimensional, noisy fractional oscillators. We also use an observation matrix $R \neq \hat{1}$, so that the mode reconstruction becomes highly nontrivial.

8.1 Numerical Set-up

The one-dimensional fractional linear oscillator is given by the following equation of motion [21]:

$$\frac{\mathrm{d}^\nu \boldsymbol{y}_a}{\mathrm{d}t^\nu} = Q_a \boldsymbol{y}_a; \quad \boldsymbol{y}_a = \begin{bmatrix} x_a \\ v_a \end{bmatrix}, \quad Q_a = \begin{bmatrix} 0 & 1 \\ -\omega_a^2 & 0 \end{bmatrix} \quad (a = 0, 1, ..., k-1), \tag{50}$$

where the frequency $\omega_a > 0$ is a real-valued parameter and $a = 0, 1, ..., k-1$ is the index of oscillators. Let us consider the multi-dimensional time-evolution equation, as follows:

$$\frac{\mathrm{d}^\nu \boldsymbol{x}}{\mathrm{d}t^\nu} = \Phi \boldsymbol{x}; \quad \boldsymbol{x} = \begin{bmatrix} \boldsymbol{y}_0 \\ \boldsymbol{y}_1 \\ \vdots \\ \boldsymbol{y}_{k-1} \end{bmatrix} \in \mathbf{R}^{2k}, \quad \Phi = \begin{bmatrix} Q_0 & & & O \\ & Q_1 & & \\ & & \ddots & \\ O & & & Q_{k-1} \end{bmatrix} \in \mathbf{R}^{2k \times 2k}, \tag{51}$$

where $\boldsymbol{y}_a$ and Q_a denote the state vector in the single-oscillator phase space and the 2×2 matrix, respectively. We also consider the observation equation as follows:

$$\boldsymbol{z}(t) = R\boldsymbol{x}(t) + \boldsymbol{\epsilon}(t), \quad \epsilon_i(t) \overset{\text{iid}}{\sim} \mathcal{N}(0, \sigma^2) \quad (i = 0, 1, ..., 2k-1), \tag{52}$$

where $R \in \mathbf{R}^{n \times 2k}$ is the constant observation matrix, and each element of $\boldsymbol{\epsilon}(t) = \left[\epsilon_0(t), \epsilon_1(t), ..., \epsilon_{k-1}(t)\right] \in \mathbf{R}^n$ is the independent and identically distributed (iid) Gaussian noise and $\boldsymbol{z} \in \mathbf{R}^n$ represents the observed signal. This numerical set-up is useful to describe the situation in which the oscillators do not interact to each other, but the resulting signal is the superposition of oscillators.

Hereafter we assume that $n \geq 2k$. The observation matrix R consists of the randomly sampled $2k$ basis vectors $v_{s(\mu)} \in \mathbf{R}^n$ $(\mu = 0, 1, ..., 2k-1)$ for a random orthonormal basis $\{v_0, v_1 ..., v_{n-1}\}$ and a permutation $s \in \mathfrak{S}_n$, as follows:

$$R = \left[v_{s(0)}, v_{s(1)}, ..., v_{s(2k-1)}\right] \in \mathbf{R}^{n \times 2k}. \tag{53}$$

Note that $s(i) \neq s(j)$ for $i \neq j$.

8.2 Numerical Tests

We perform the numerical tests to determine the frequency ω_a $(a = 0, 1, ..., k-1)$ in Eq. (50) and Eq. (51). The eigenvalues of the coefficient matrix Φ are $\pm\sqrt{-1}\omega_a$ $(a = 0, 1, ..., k-1)$. We also denote the eigenvalues of the coefficient matrix A obtained by the fracDMD (without control) by λ_i $(i = 0, 1, ..., 2k-1)$. Let us assume that $\omega_0 > \omega_1 > ... > \omega_{k-1} > 0$ and $\operatorname{Im}\lambda_0 > \operatorname{Im}\lambda_1 > ... > \operatorname{Im}\lambda_{k-1} > \operatorname{Im}\lambda_{2k-1} > \operatorname{Im}\lambda_{2k-2} > ... > \operatorname{Im}\lambda_k$ for simplicity.

The error L of the frequency estimation is now given by the followings:

$$L = \sum_{i=0}^{k-1} |\lambda_i - \omega_i|^2 + \sum_{i=0}^{k-1} |\lambda_{i+k} + \omega_i|^2. \tag{54}$$

In the full reconstruction case, $\lambda_i = \sqrt{-1}\omega_i$ and $\lambda_{k+i} = -\sqrt{-1}\omega_i$ $(i = 0, 1, ..., k-1)$, so that $L = 0$.

We show the frequency reconstruction error L for various values of noise standard deviation σ and the dimension of the observation vectors n in Fig. 1 below. We assume that the rank $2k$ and the order ν of the system equation Eq. (51) are known. We use the second-order numerical discretization of the Caputo fractional differential, instead of the first-order scheme shown in Eq. (41) and Eq. (42). We also modified Eq. (44) so that we can use the fractional *integral* instead of the fractional differential with positive order α in order to achieve a better numerical convergence. For the details of the numerical implementations, see Supplementary Materials.

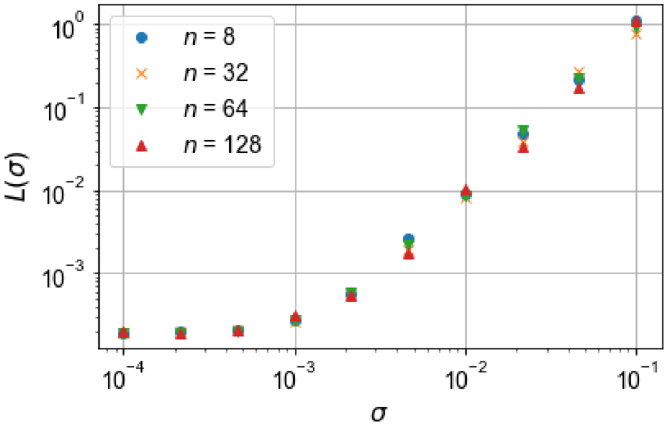
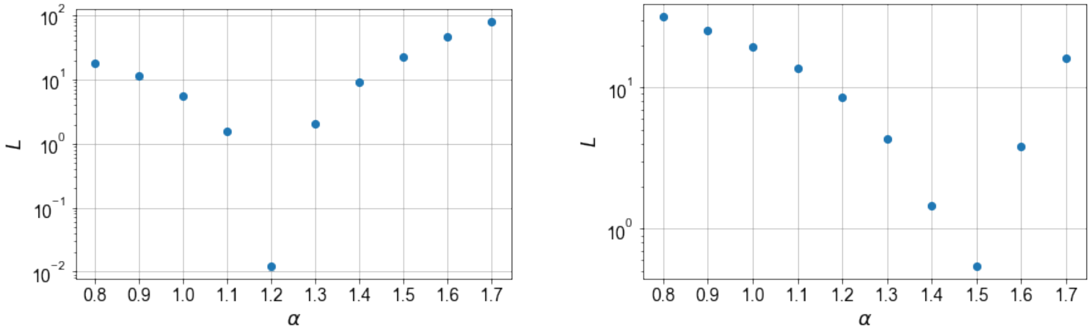


Figure 1: The frequency reconstruction error L against the noise standard deviation σ, for various values of observation size n. The numerical set-ups are as follows: $k = 4$, $\omega_i = i + 1$ $(i = 0, 1, 2, 3)$, and the order of the system equation Eq. (51) is set to $\nu = 1.2$. The fracDMD parameters are as follows: the SVD rank r is set to the actual system size $2k$, and the order α is set to the actual value ν. Each mark denotes a mean of 10 synthetic data generated by the system equation with different noise realizations and initial conditions. The initial conditions for each oscillator are randomly chosen so that the initial (pseudo) energy of each oscillator becomes unity.

We can see that the reconstruction error L is an increasing function of σ for $\sigma > 10^{-3}$, while L is almost constant for $\sigma < 10^{-3}$. This implies that major part of the reconstruction error vanishes with $\sigma \to 0$, leaving the constant part of the error for small σ.



(a) $\nu = 1.2$

(b) $\nu = 1.5$

Figure 2: The frequency reconstruction error L against the order α used in the fracDMD. (a) $\nu = 1.2$, (b) $\nu = 1.5$. In both tests, $\sigma = 10^{-2}$ and $n = 8$. The other numerical set-ups are the same as in Fig. 1. The fracDMD parameters are as follows: the SVD rank r is set to the actual system size $2k$.

As we can see in Fig. 2, the function $L(\alpha)$ has with its minima located around $\alpha = \nu$, indicating that a *wrong* value of α leads to a larger value of reconstruction error.

9 Discussion

The fractional oscillator is a model for the oscillator with power-law memory effects. The reconstruction of the frequencies in the previous section is in a good agreement with the ground truth. The reconstruction error increases with the increasing noise standard deviations, while the dimension of the observation space does not affect the error severely. We thus conclude that the fracDMD can reconstruct the isolated fractional oscillators well.

We also performed numerical experiments for various values of α. It is shown that the reconstruction error L has its minima around $\alpha = \nu$. This implies that our method is also useful to estimate the order of the system equation. Noting that the case $\alpha = 1$ roughly corresponds to the DMD, we can also infer that our method achieves a better reconstruction than the DMD for the fractional oscillators. Although this inference seems exact, the introduction of a higher-order scheme and the fractional integral might lead to a small discrepancy between the DMD and the fracDMD results even for $\alpha = 1$.

Our method can be used to model unknown physical processes e.g., mechanical motion of industrial machines and thermal systems. One possible way to use the method is to estimate the memory effect in the system. If the reconstruction error by the fracDMD has its minima around $\alpha = 1$, one may conclude that the system is memoryless; otherwise, one can use the optimal value of α to include the memory effects to the model.

Among possible extensions, nonlinear coupled oscillators are one possible way to extend our numerical tests. Another way to include more complex situations is to use the non power-law memories. In our current numerical set-up, we use power-law memories, which is mathematically shown to be equivalent to the fractional-order equations of motion [21]. However, the memories in real data can, in principle, decay according to any smooth functions.

In the reconstruction considered in the previous section, we assume that we know the actual size of the problem (i.e., the system size). This assumption makes the problem easy to approach. In a realistic situation, however, the actual dimension of the system equation is seldom known. In a future work, one may optimize the rank of the system as well as the coefficients.

10 Summary

We proposed a new framework, DMD with memory, or, DMDm, a DMD-based numerical tool to analyze time-series data. Use of more generic linear operator instead of the finite difference operator enables us to take the memory effects into account in the time-evolution models. The memory effect is a widely seen phenomena in real world, e.g., in a viscoelastic matter and in fluid dynamics. As one example of the DMDm, we formulated the fracDMD, whose name stands for *fractional* DMD, indicating the use of fractional-order derivative. The incorporation of fractional-order derivative in DMDm is equivalent to assume the power-law memory effects. We successfully show that using the fracDMD one can reconstruct the frequencies of fractional oscillators from noisy observations. The proposed method is expected to be useful to model unknown physical processes, such as thermal and mechanical processes in modern industrial machines.

Author Contributions

RA developed the proposed methods, formulated the DMDm and the fracDMD, and designed the numerical experiments. KS developed the numerical schemes (including the higher-order scheme for the fracDMD) and built the numerical codes. Both of them performed numerical tests in Python and discussed on the results. TT, MK, and TM supervised the project and gave comments.

Conflict of Interest

We declare that we have no conflicts of interest.

Acknowledgments

We would like to thank our teammates for their comments on the theoretical and practical aspects of our methods. Mr. Takashi Yokota and Mr. Masaru Nishimura of Tokyo Electron Ltd. gave us valuable comments on the novelty of the proposed methods. We also thank their aids for patent applications for the proposed methods. Our survey of existing research works is partly helped by the Access to Articles program in Tokyo Electron Group.

Supplementary Materials

A Improvements in Numerical Implementation

We introduce the fracDMD (fractional dynamic mode decomposition) using the first-order discretization of the Caputo derivative in the main text. In this section, we show two ingredients to improve the numerical scheme for the fracDMD.

In appendix A.2, we show a scheme the discretization error converges to zero with the second-order in time intervals. One can expect more accurate estimations of the coefficient matrices for the second-order scheme. In appendix A.3, we show the use of fractional *intengral*, instead of the fractional derivative in the fracDMD. In the fractional calculus, one might think that the derivative and integral are treated in an unified manner; however, there are some subtleties in the fractional derivative, because the derivative of order $\alpha > 1$ requires the integer-order derivatives in the Caputo derivative. For some limited cases, we can circumvent the use of fractional derivatives and hence integer-order derivatives by *integrating* both sides of the time-evolution equation.

In this section, we assume a set of time points $\mathsf{T} = \{t_i = i\Delta t | i = 0, 1, ..., m-1\}$ for $\Delta t > 0$.

A.1 First-order numerical scheme revisited

We would like to discretize the following equation within a *first-order* error in Δt, assuming $T = t_{m-1} - t_0$ is a constant. For a smooth integrable function $f : \mathbf{C} \to \mathbf{C}$, the α-th order fractional derivative at time t is derived as follows:

$$D_f^{(\alpha)}(t) = \frac{1}{\Gamma(\alpha)} \int_0^t \mathrm{d}\tau \, f(\tau)(t-\tau)^{\alpha-1}. \tag{55}$$

Noting that $f(\tau) = f(t_i) + \mathcal{O}(\Delta t)$ for $\tau \in [t_i, t_{i+1}]$, the above expression for $t = t_{k+1}$ becomes,

$$\begin{aligned} &D_f^{(\alpha)}(t_{k+1}) \\ &= \frac{1}{\Gamma(\alpha)} \sum_{i=0}^{k} \int_{t_i}^{t_{i+1}} \mathrm{d}\tau \, (f(t_i) + \mathcal{O}(\Delta t))(t_{k+1} - \tau)^{\alpha-1} \\ &= \frac{1}{\Gamma(\alpha)\alpha} \sum_{i=0}^{k} f(t_i)((t_{k+1} - t_i)^{\alpha} - (t_{k+1} - t_{i+1})^{\alpha}) + \mathcal{O}(\Delta t), \end{aligned} \tag{56}$$

where we use the fact $\mathcal{O}(k\Delta t^2) = \mathcal{O}(\Delta t)$ for $k \simeq m$. Hence,

$$D(t_{k+1}) = \frac{1}{\Gamma(\alpha+1)} \sum_{i=0}^{k} ((t_{k+1} - t_i)^{\alpha} - (t_{k+1} - t_{i+1})^{\alpha}) \, f(t_i) + \mathcal{O}(\Delta t). \tag{57}$$

In the matrix representation Eq. (41) in the main text, we want the matrix $W^{(\alpha)} = \left[\boldsymbol{w}_0^{(\alpha)}, \boldsymbol{w}_1^{(\alpha)}, ..., \boldsymbol{w}_{m-1}^{(\alpha)}\right]$ to be invertible. To meet this end, we can simply store the derivative at $t = t_{k+1}$ in the k-th element of the resultant (row) vector $\vec{y} = \vec{x}W^{(\alpha)}$ for a given time-series data $\vec{x}$ so that the diagonal elements of $W^{(\alpha)}$ become nonzero, without violating the requirements for the first-order scheme. Adopting this convention, we can derive the expression for the weights Eq. (42) in the main text.

A.2 Second-order numerical scheme

In this subsection, we derive a second-order discretization scheme for the Caputo derivative. We perform a discretization of the following integral:

$$D_f^{(\nu)}(t) = \frac{1}{\Gamma(\nu)} \int_0^t \mathrm{d}\tau \, \frac{f(\tau)}{(t-\tau)^{1-\nu}}, \tag{58}$$

where the integrand $f : \mathbf{C} \to \mathbf{C}$ is a integrable scalar function and $\nu < 1$ is the real-valued order. The value of $D_f^{(\nu)}(t)$ is evaluated as

$$D_f^{(\nu)}(t_{k+1}) = D_f^{(\nu)}((k+1)\Delta t) = \frac{1}{\Gamma(\nu)} \int_0^{(k+1)\Delta t} \mathrm{d}\tau \, \frac{f(\tau)}{((k+1)\,\Delta t - \tau)^{1-\nu}}. \tag{59}$$

Let us define $\mathcal{F}_k = \{f_0, f_1, ..., f_{k-1} | f_i = f(t_i)\}$ for $k = 1, 2, ..., m$. We would like to approximate $D_f^{(\nu)}(t_{k+1})$ $(0 < k < m-1)$ using $\mathcal{F}_{k+1}$. A direct calculation of $D_f^{(\nu)}(t_{k+1})$ is as follows:

$$\begin{aligned} &\frac{1}{\Gamma(\nu)} \int_0^{(k+1)\Delta t} \mathrm{d}\tau \, \frac{f(\tau)}{((k+1)\,\Delta t - \tau)^{1-\nu}} \\ &= \frac{1}{\Gamma(\nu)} \sum_{i=0}^{k} \int_{i\Delta t}^{(i+1)\Delta t} \mathrm{d}\tau \, \frac{f(\tau)}{((k+1)\Delta t - \tau)^{1-\nu}} \\ &= \frac{1}{\Gamma(\nu)} \sum_{i=0}^{k} \int_0^1 \mathrm{d}s \, \Delta t \frac{f(i\Delta t + s\Delta t)}{((k+1)\Delta t - (i\Delta t + s\Delta t))^{1-\nu}} \\ &= \frac{1}{\Gamma(\nu)} \sum_{i=0}^{k} \int_0^1 \mathrm{d}s \, \Delta t \, ((k+1)\Delta t - (i+s)\Delta t)^{\nu-1} \, f(i\Delta t + s\Delta t). \end{aligned} \tag{60}$$

Assuming that f is sufficiently smooth, we can use the Taylor expansion to get the following approximation for $0 < s < 1$.

$$f(i\Delta t + s\Delta t) = (1-s) f_i + s f_{i+1} + \mathcal{O}(\Delta t^2). \tag{61}$$

Substituting Eq. (61) into Eq. (60), we obtain the following expression for $D_f^{(\nu)}(t_{k+1})$.

$$D_f^{(\nu)}(t_{k+1}) = \frac{1}{\Gamma(\nu)} \sum_{i=0}^{k} \int_0^1 \mathrm{d}s \, (\Delta t)^{\nu} ((k+1) - (i+s))^{\nu-1} [(1-s) f_i + s f_{i+1}] + \mathcal{O}(\Delta t^2). \tag{62}$$

Let us introduce the *distance* between time points k, i by $C_{k,i} = k - i + 1$ and normalized (inverse) distance $D_{k,i} = 1 - C_{k,i}^{-1} \in [0, 1)$. A straight-forward calculation yields

$$\begin{aligned} D_f^{(\nu)}(t_{k+1}) = &\frac{1}{\Gamma(\nu+2)} \sum_{i=0}^{k} \Delta t^{\nu} C_{k,i}^{\nu+1} \left[f_i \left(D_{k,i}^{\nu+1} - D_{k,i} - \nu C_{k,i}^{-1} \right) + f_{i+1} \left(1 - D_{k,i}^{\nu} \left(1 + \nu C_{k,i}^{-1} \right) \right) \right] \\ &+ \mathcal{O}(\Delta t^2). \end{aligned} \tag{63}$$

We want to express the summation in Eq. (63) using dot products of constant weight vectors. Let $\boldsymbol{\psi}^{(k+1)} = \left[\psi_0^{(k+1)}, \psi_1^{(k+1)}, ... \psi_{k+1}^{(k+1)} \right]^{\top} \in \mathbf{R}^{k+2}$ denote the weight vector:

$$\psi_i^{(k+1)} = \begin{cases} P_{k+1}^{(\nu)}(0) & (i = 0) \\ P_{k+1}^{(\nu)}(i) + Q_{k+1}^{(\nu)}(i-1) & (1 \le i \le k) \\ Q_{k+1}^{(\nu)}(k) & (i = k+1), \end{cases} \tag{64}$$

where,

$$\begin{aligned} P_{k+1}^{(\nu)}(i) &= \frac{1}{\Gamma(\nu+2)} \Delta t^{\nu} C_{k,i}^{\nu+1} \left(D_{k,i}^{\nu+1} - D_{k,i} - \nu C_{k,i}^{-1} \right), \\ Q_{k+1}^{(\nu)}(i) &= \frac{1}{\Gamma(\nu+2)} \Delta t^{\nu} C_{k,i}^{\nu+1} \left[1 - D_{k,i}^{\nu} \left(1 + \nu C_{k,i}^{-1} \right) \right]. \end{aligned} \tag{65}$$

Now that, the value of the fractional derivative $D_f^{(\nu)}$ at time $t = t_{k+1}$ is estimated by the following expression.

$$D_f^{(\nu)}(t_{k+1}) = \big[f_0, f_1, ..., f_{k+1}\big]\boldsymbol{\psi}^{(k+1)} + \mathcal{O}(\Delta t^2). \tag{66}$$

Note that Eq. (66) is the second-order scheme for Eq. (58). To implement the second-order numerical scheme for the fracDMD, one may simply replace the weight vector $\boldsymbol{w}_k^{(\nu)}$ in Eq. (42) by the following expression:

$$(\boldsymbol{w}_k^{(\nu)})_i = \begin{cases} \psi_i^{(k)} & (i \leq k) \\ 0 & (i > k). \end{cases} \tag{67}$$

The vanishing elements for $i > k$ correspond to the causality in the time-evolution equation.

A.3 Use of the fractional integral

In the main text, we derive the fracDMD using the fractional differential equation Eq. (46) in the main text, in which the fractional derivative of the time-dependent variable $\boldsymbol{x} \in \mathbf{R}^n$ is expressed in terms of a linear function of $\boldsymbol{x}$ and the exogenous input term $\boldsymbol{u}(t) \in \mathbf{R}^\ell$, as follows:

$$\frac{\mathrm{d}^\alpha \boldsymbol{x}}{\mathrm{d}t^\alpha} = A\boldsymbol{x} + B\boldsymbol{u}(t). \tag{68}$$

However, a naïve discretization of the original form Eq. (68) leads to a subtlety in numerical treatments around $t = 0$: The solution of the time-evolution equation Eq. (46) is, in general, a nonzero vector at $t = 0$. However, the Caputo derivative of a function vanishes at the initial time $t = 0$, leading to the inconsistency at $t = 0$. This is the major reason why we assumed that $\pi \in \mathrm{hom}(\mathbf{C}^{\mathcal{T}}, \mathbf{C}^{\mathcal{S}})$ with $\mathcal{S} = (0, T] \subset [0, T] = \mathcal{T}$ for the linear operator π in the main text.

Alternatively, to circumvent the difficulty, one might think that we could perform a fractional derivative of order $-\alpha$ to both sides of the equation Eq. (68) to *cancel* the derivative in the left-hand side, and get an alternative form for $\alpha < 0$ as,

$$\boldsymbol{x} \overset{?}{=} \frac{\mathrm{d}^{-\alpha}}{\mathrm{d}t^{-\alpha}}(A\boldsymbol{x} + B\boldsymbol{u}(t)). \tag{69}$$

We can show that this transformation is exact for $\alpha < 0$ [5]. For an arbitrary integrable function $\boldsymbol{x} = \boldsymbol{x}(t)$ and $\alpha > 0$, however, the transformation from the original form Eq. (68) to the alternative form Eq. (69) is not possible because the function $\boldsymbol{x}$ has nonzero integer-order derivatives at $t = 0$ in general. However, one can show that the difference between the original function $\boldsymbol{x}$ and the retrieved function $\tilde{\boldsymbol{x}} = \frac{\mathrm{d}^{-\alpha}}{\mathrm{d}t^{-\alpha}}\frac{\mathrm{d}^\alpha \boldsymbol{x}}{\mathrm{d}t^\alpha}$ is expressed in a polynomial of t, as follows (see Lemma 2.22 in [5]):

$$\frac{\mathrm{d}^{-\alpha}}{\mathrm{d}t^{-\alpha}}\frac{\mathrm{d}^\alpha \boldsymbol{x}}{\mathrm{d}t^\alpha} = \boldsymbol{x}(t) - \sum_{\ell=0}^{n_\alpha^+ - 1} \frac{\boldsymbol{x}^{(\ell)}(0)}{\ell!} t^\ell. \tag{70}$$

Noting that $\alpha = 0$ is a trivial case, we can perform the mode decomposition by discretizing the alternative form Eq. (69) for $\alpha < 1$. The resultant equation becomes as follows:

$$\boldsymbol{x} = \frac{\mathrm{d}^{-\alpha}}{\mathrm{d}t^{-\alpha}}(A\boldsymbol{x} + B\boldsymbol{u}(t)) + \boldsymbol{x}(0). \tag{71}$$

Even for $\alpha > 1$, Eq. (71) serves as an approximation for the exact fractional integral equation Eq. (70). One can derive an alternative version of the fracDMD using Eq. (71) through the same procedure as we have seen in the main text.

B Numerical Test

In Section 8, we use the approximated alternative form of the fracDMD shown in the previous section (i.e., the fractional integral equation Eq. (71) and the second-order discretization Eq. (67)).

References

[1] T. Askham and J. N. Kutz, *Variable projection methods for an optimized dynamic mode decomposition*, SIAM Journal on Applied Dynamical Systems, 17 (2018), pp. 380–416.

[2] S. L. Brunton, M. Budišić, E. Kaiser, and J. N. Kutz, *Modern koopman theory for dynamical systems*, arXiv preprint arXiv:2102.12086, (2021).

[3] S. Burov and E. Barkai, *Fractional langevin equation: Overdamped, underdamped, and critical behaviors*, Physical Review E, 78 (2008), p. 031112.

[4] H. J. Haubold, A. M. Mathai, and R. K. Saxena, *Mittag-leffler functions and their applications*, Journal of applied mathematics, 2011 (2011).

[5] A. A. Kilbas, H. M. Srivastava, and J. J. Trujillo, *Theory and applications of fractional differential equations*, vol. 204, elsevier, 2006.

[6] D. Matignon, *Stability results for fractional differential equations with applications to control processing*, in Computational engineering in systems applications, vol. 2, Citeseer, 1996, pp. 963–968.

[7] I. Mezić, *Spectral properties of dynamical systems, model reduction and decompositions*, Nonlinear Dynamics, 41 (2005), pp. 309–325.

[8] I. Mezic, *Analysis of fluid flows via spectral properties of the koopman operator*, Annual Review of Fluid Mechanics, 45 (2013), pp. 357–378.

[9] I. Podlubny, *Fractional-order systems and pi/sup /spl lambda//d/sup /spl mu//-controllers*, IEEE Transactions on Automatic Control, 44 (1999), pp. 208–214, https://doi.org/10.1109/9.739144.

[10] J. L. Proctor, S. L. Brunton, and J. N. Kutz, *Dynamic mode decomposition with control*, SIAM Journal on Applied Dynamical Systems, 15 (2016), pp. 142–161.

[11] E. Rodrigues, B. Zadrozny, C. Watson, and D. Gold, *Decadal forecasts with resdmd: a residual dmd neural network*, arXiv preprint arXiv:2106.11111, (2021).

[12] C. W. Rowley, I. Mezić, S. Bagheri, P. Schlatter, and D. S. Henningson, *Spectral analysis of nonlinear flows*, Journal of fluid mechanics, 641 (2009), pp. 115–127.

[13] I. Sakata, T. Sakata, K. Mizoguchi, S. Tanaka, G. Oohata, I. Akai, Y. Igarashi, Y. Nagano, and M. Okada, *Complex energies of the coherent longitudinal optical phonon–plasmon coupled mode according to dynamic mode decomposition analysis*, Scientific Reports, 11 (2021), pp. 1–10.

[14] A. A. K. Samko, Stefan G. and O. I. Marichev, *Fractional integrals and derivatives: theory and applications*, Gordon and Breach Science Publishers, 1993.

[15] M. Sasaki, Y. Kawachi, R. Dendy, H. Arakawa, N. Kasuya, F. Kin, K. Yamasaki, and S. Inagaki, *Using dynamical mode decomposition to extract the limit cycle dynamics of modulated turbulence in a plasma simulation*, Plasma Physics and Controlled Fusion, 61 (2019), p. 112001.

[16] D. Sashidhar and J. N. Kutz, *Bagging, optimized dynamic mode decomposition for robust, stable forecasting with spatial and temporal uncertainty quantification*, Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, 380 (2022), p. 20210199, `https://doi.org/10.1098/rsta.2021.0199`.

[17] P. J. Schmid, *Dynamic mode decomposition of numerical and experimental data*, Journal of fluid mechanics, 656 (2010), pp. 5–28.

[18] P. J. Schmid, *Dynamic mode decomposition and its variants*, Annual Review of Fluid Mechanics, 54 (2022), pp. 225–254, `https://doi.org/10.1146/annurev-fluid-030121-015835`.

[19] A. A. Stanislavsky, *Fractional oscillator*, Phys. Rev. E, 70 (2004), p. 051103, `https://doi.org/10.1103/PhysRevE.70.051103`, `https://link.aps.org/doi/10.1103/PhysRevE.70.051103`.

[20] N. Sugimoto and T. Horioka, *Dispersion characteristics of sound waves in a tunnel with an array of helmholtz resonators*, Journal of the Acoustical Society of America, 97 (1995), pp. 1446–1459.

[21] A. Svenkeson, B. Glaz, S. Stanton, and B. J. West, *Spectral decomposition of nonlinear systems with memory*, Phys. Rev. E, 93 (2016), p. 022211, `https://doi.org/10.1103/PhysRevE.93.022211`.

[22] V. E. Tarasov, *General fractional dynamics*, Mathematics, 9 (2021), p. 1464.

[23] J. Tenreiro Machado and A. Azenha, *Fractional-order hybrid control of robot manipulators*, in SMC'98 Conference Proceedings. 1998 IEEE International Conference on Systems, Man, and Cybernetics (Cat. No.98CH36218), vol. 1, 1998, pp. 788–793 vol.1, `https://doi.org/10.1109/ICSMC.1998.725510`.

[24] 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, 1 (2014), pp. 391–421.