Dynamic Mode Decomposition with Memory

Ryoji Anzaki,[1,*] Kei Sano,[2,†] Takuro Tsutsui,[3,‡] Masato Kazui,[2,§] and Takahito Matsuzawa[2,¶]

[1] *Advanced Engineering 1st Department, Digital Design Center, Tokyo Electron Ltd., Akasaka Biz Tower, 3-1 Akasaka 5-chome, Minato-ku, Tokyo 107-6325, Japan*

[2] *Advanced Engineering 1st Department, Digital Design Center, Tokyo Electron Ltd., Daido Seimei Sapporo Building, 1, Kita 3-jo, Nishi 3-chome, Chuo-ku, Sapporo City, Hokkaido, 060-0003, Japan*

[3] *Advanced Engineering 2nd Department, Digital Design Center, Tokyo Electron Ltd., Daido Seimei Sapporo Building, 1, Kita 3-jo, Nishi 3-chome, Chuo-ku, Sapporo City, Hokkaido, 060-0003, Japan*

(Dated: August 8, 2023)

This study proposed a novel method of dynamic mode decomposition with memory (DMDm) to analyze multi-dimensional time-series data with memory effects. The memory effect is a widely observed phenomenon in physics and engineering and is considered to be the result of interactions between the system and environment. Dynamic mode decomposition (DMD) is a linear operation-based, data-driven method for multi-dimensional time-series data proposed in 2008. Although DMD is a successful method for time-series data analysis, it is based on ordinary differential equations and thus, cannot incorporate memory effects. In this study, we formulated the abstract algorithmic structure of DMDm and demonstrate its utility in overcoming the *memoryless* restriction imposed by existing DMD methods on the time-evolution model. In the numerical demonstration, we utilized the Caputo fractional differential to implement an example of DMDm such that the time-series data could be analyzed with power-law memory effects. Thus, we developed a fractional DMD, which is a DMD-based method with arbitrary (real value) order differential operations. The proposed method was applied to synthetic data from a set of fractional oscillators and model parameters were estimated successfully. The proposed method is expected to be useful for scientific applications, and aid in model estimation, control, and failure detection of mechanical, thermal, and fluid systems in factory machines, such as modern semiconductor manufacturing equipment.

I. INTRODUCTION

Dynamic mode decomposition (DMD) [1–4] is a data-driven, linear algebra-based method for time-series data analysis. It was first developed by Schmid for the analysis of experimental data in fluid dynamics [1] and is now widely used in a wide variety of scientific fields such as climatology [5, 6], plasma physics [7], dissipative quantum systems [8], and fluid dynamics applications [9]. In addition to its success in data-driven science, the mathematical structure of DMD has garnered attention, particularly in terms of its connection with Koopman theory [3, 10, 11].

The basic idea of DMD is to find the best-fit matrix coefficient assuming a constant-coefficient linear difference equation for a given multi-dimensional time-series data. Using DMD, we can extract the coefficients of the time-evolution equation and time-dependent modes of the dynamics. In DMD, a mode varies both in spatial- and temporal directions, and that is the reason of the name *dynamic mode* decomposition. For example, in [7], Sasaki *et al.* extracted the time-evolution patterns of the plasma using DMD.

* ryoji.anzaki@tel.com
† kei.sano@tel.com
‡ takuro.tsutsui@tel.com
§ masato.kazui@tel.com
¶ matty.matsuzawa@tel.com

A. DMD Algorithm

We show the DMD algorithm in its simplest form, *exact* DMD [2, 4]. Consider a set of time points $\mathsf{T} = \{t_0, t_1, ..., t_{m-1}\}$ with $t_j > t_i$ for $j > i$. Assume 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 a 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, as follows:

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

where $\mathcal{A} \in \mathbf{R}^{n\times n}$ is a coefficient matrix of the continuous time-evolution model. For a uniformly discretized T with time interval $\Delta t > 0$, $\mathcal{A}$ is estimated by 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 follows:

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

where $\hat{1}_n$ is the $n \times n$ unit matrix and $\| \bullet \|$ is the matrix Frobenius norm. 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 using 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; that is, $U^*U = \hat{1}_n$ and

$V^*V = \hat{1}_m$, where $\bullet^*$ denotes the adjoint matrix. In this paper, we use $K^{a\times b}$ to denote the set of $a \times b$ matrices with their matrix elements being in $K = \mathbf{R}, \mathbf{C}$. Matrix $\Sigma \in \mathbf{R}^{n\times m}$ contains singular values of X. Note that we use the *full* SVD, i.e., Σ is an oblong matrix, whereas U and V are square matrices, so that the equalities $UU^* = \hat{1}_n = U^*U$ and $VV^* = \hat{1}_m = V^*V$ holds exactly. The pseudo-inverse X^+ is now calculated as $X^+ = V\Sigma^+U^*$. Consequently, the parameter of the discrete-time-evolution model is expressed as

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

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 obtain the rank-r representation of the discrete-time time-evolution model as follows:

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

Low-rank representations are frequently used for data with large dimensions.

B. 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., as the eigenvectors of the coefficient matrix $\mathcal{A}$) with the time-evolution function expressed as a time-dependent exponential function:

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

where $\boldsymbol{\Phi} = \left[\vec{\phi}_0, \vec{\phi}_1, ..., \vec{\phi}_{n-1}\right]$ and $\boldsymbol{\Omega} = \mathrm{diag}(\omega_0, \omega_1, ..., \omega_{n-1})$ are 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, and b_j is the loading for each DMD mode. The DMD mode and the corresponding exponential function have the same eigenvalue to satisfy Eq. (1). A low-rank representation corresponds to the replacement of $\sum_{j=0}^{n-1} \to \sum_{j=0}^{r-1}$ in Eq. (5) for $1 < r < n$.

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

C. Non-uniform Time Points

A set of time points T can be uneven in several realistic situations. In these cases, a discretized representation of the time-derivative operator can be constructed. Hereafter, we assume that data matrix X is defined by $X = [\boldsymbol{x}_0, \boldsymbol{x}_1, ..., \boldsymbol{x}_{m-1}]$. The time-evolution model now has the form

$$XD_m^{(1)} = A\big[\mathbf{0}, \boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_{m-1}\big], \tag{6}$$

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

$$D_m^{(1)} = \begin{bmatrix} 0 & -\theta_{01} & 0 & & 0 \\ 0 & +\theta_{01} & -\theta_{12} & & 0 \\ 0 & 0 & +\theta_{12} & \ddots & 0 \\ \vdots & & & \ddots & -\theta_{m-2,m-1} \\ 0 & 0 & 0 & & +\theta_{m-2,m-1} \end{bmatrix}, \tag{7}$$

where $\theta_{k-1,k} = 1/(t_k - t_{k-1}) > 0$. Eq. (6) exhibits itself as a special case of the following generic matrix equation for an integer $0 \le q \le m-1$ and an upper triangular matrix $D \in \mathbf{C}^{m\times m}$ whose first q columns are zero vector $\mathbf{0} = \big[0, ..., 0\big]^\top$:

$$XD = A\big[\mathbf{0}, ..., \mathbf{0}, \boldsymbol{x}_q, \boldsymbol{x}_{q+1}, ..., \boldsymbol{x}_{m-1}\big], \tag{8}$$

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

D. DMD with Control

Among the extensions of DMD, DMD with control (DMDc) [12] is a method used for analyzing time-series data corresponding to a non-autonomous dynamical system. The term *non-autonomous* refers to the existence of an exogenous external force, whose time dependence is not affected by the status of the dynamical system itself. One example is the forced oscillation, wherein an external force is applied to the oscillator. The generic form of the non-autonomous dynamical system of interest is expressed as follows:

$$\frac{\mathrm{d}\boldsymbol{x}}{\mathrm{d}t} = A\boldsymbol{x}(t) + B\boldsymbol{u}(t), \tag{9}$$

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

$$\big[A\ B\big] \simeq \big[\bar{A}\ \bar{B}\big] = XD_m^{(1)} \begin{bmatrix} \mathbf{0}, \boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_{m-1} \\ \mathbf{0}, \boldsymbol{u}_1, \boldsymbol{u}_2, ..., \boldsymbol{u}_{m-1} \end{bmatrix}^+, \tag{10}$$

where $[A\ B]$ is a block matrix with two blocks A and B. The application of the standard procedure for singular-value decomposition leads to reduced representation [12].

E. Other Related Works

The simple, highly extensive algorithmic structure of DMD has aided researchers in developing better numerical methods based on the original DMD. Residual DMD (resDMD) [5] is a neural-network-based method with residual blocks, and its reference is replaced by the results of the DMD algorithm. The incorporation of neural networks enables resDMD to handle time-series data that are best fitted by highly nonlinear time-evolution models. The optimized DMD (optDMD) [13] and bagging, optimized DMD (BOP–DMD) [6] are extensions of DMD, with their nonlinear optimization yielding better de-biasing outcomes. Conversely, the bagging of snapshots provides a better convergence to the optimization and enables uncertainty quantification.

F. Generic Time-Evolution Equation

In this paper, we define *time-evolution* as an initial-value problem (IVP). An IVP is a pair of initial conditions and time-evolution equation. For instance, for $\ell \in \mathbf{Z}_{>0}$, a ℓ-th order ODE $\frac{\mathrm{d}^\ell \boldsymbol{x}}{\mathrm{d}t^\ell} = f(\boldsymbol{x}(t), t),\ \boldsymbol{x} \in \mathbf{R}^n$ with ℓ initial conditions $\boldsymbol{x}(0) = \boldsymbol{x}_{\mathrm{ini},0}, ..., \boldsymbol{x}^{(\ell-1)}(0) = \boldsymbol{x}_{\mathrm{ini},\ell-1}$ is an IVP. The number of independent initial conditions q necessary to specify the time-series $\boldsymbol{x}(t)$ depends on the time-evolution equation. Let us call the number of initial conditions q the *nullity* of the time-evolution equation or the corresponding operator in the time domain. In the example above, q is equal to the order of the ODE. This generic idea can be further generalized to other types of time-evolution equations, such as difference equation in the discrete cases. In this paper, we concentrate on the IVPs, leaving other ways to specify the degrees of freedom (e.g., final-value problem) for the future works.

In general, the time-evolution equation on the set of time points $\mathcal{T}$ has the form

$$\pi(\boldsymbol{x})(t) = f(\boldsymbol{x}(t), t),\ \boldsymbol{x} \in \mathbf{R}^n,\ t \in \mathcal{T}, \tag{11}$$

where π is a functional of $\boldsymbol{x}$, corresponding to the transformation of $\boldsymbol{x}$ in time domain. To interpret Eq. (11) as a time-evolution equation, one has to impose the *causality* on the functional π: $\pi(\boldsymbol{x})(t)$ is *not* dependent on $\boldsymbol{x}(t')$ for any $t' > t$ $(t, t' \in \mathcal{T})$. If the functional π is not causal, Eq. (11) together with initial conditions falls into a self-reference, and thus irrelevant for a time-evolution function.

Note that, in a functional space with sufficiently smooth functions, a differential operator $\frac{\mathrm{d}^\ell}{\mathrm{d}t^\ell}$ has causality, since it is equivalent to the *left* derivative, and thus can be calculated without using the *future* states. Let us suppose that the set of time points is an interval, i.e., $\mathcal{T} = [0, T]$ with $T > 0$. At the minimum of the interval $t = 0$, a function has no left derivative, and thus the domain of the differential operator $\frac{\mathrm{d}^\ell}{\mathrm{d}t^\ell}$ is $(0, T]$. The lack of the derivatives at $t = 0$ is the origin of the nonzero nullity of the ODEs as time-evolution equations.

Let us discretize the interval $[0, T]$ into m time points $t_i = i\Delta t$ for $i = 0, ..., m-1$ with $\Delta t = T/(m-1)$. Then we can construct a discretized representation of the ℓ-th order ODE as follows:

$$\begin{aligned} &D_m^{(\ell)} \left[\boldsymbol{x}(t_0), \boldsymbol{x}(t_1), ..., \boldsymbol{x}(t_{m-1})\right] \\ &\qquad = \left[\mathbf{0}, ..., \mathbf{0}, f(\boldsymbol{x}(t_\ell), t_\ell), ..., f(\boldsymbol{x}(t_{m-1}), t_{m-1})\right], \end{aligned} \tag{12}$$

where $D_m^{(\ell)}$ is a ℓ-th order difference operator. For $\ell = 1$, one can find $D_m^{(1)}$ in Eq. (7). Note that, the rank of the difference matrix is given by $\mathrm{rank}(D_m^{(\ell)}) = m - \ell$, and thus our choice of the term *nullity* coincides with that in linear algebra (i.e., dimension minus rank for a square matrix). To specify the vectors $\boldsymbol{x}(t_i)$ for $i = 0, 1, ..., m-1$, one needs $q = m - \ell$ independent initial conditions, such as

$$\begin{aligned} &\boldsymbol{x}(t_0) = \boldsymbol{c}_0,\ \boldsymbol{x}(t_1) - \boldsymbol{x}(t_0) = \Delta t \boldsymbol{c}_1, \\ &\boldsymbol{x}(t_2) - 2\boldsymbol{x}(t_1) + \boldsymbol{x}(t_0) = \Delta t^2 \boldsymbol{c}_2\ \ldots, \end{aligned} \tag{13}$$

For given constant vectors $\boldsymbol{c}_0, \boldsymbol{c}_1, \boldsymbol{c}_2, ...$, corresponding to the *position*, *velocity*, and *acceleration* at $t = 0$. In general, one can specify the initial conditions by the following linear equation:

$$\left[\boldsymbol{x}(t_0), \boldsymbol{x}(t_1), ..., \boldsymbol{x}(t_{q-1})\right] \Omega = \boldsymbol{\gamma}, \tag{14}$$

for given constant vector $\boldsymbol{\gamma} \in \mathbf{R}^q$ and an invertible matrix $\Omega \in \mathbf{R}^{q \times q}$.

G. Memory Effects

Despite the tremendous success of the aforementioned methods, there is scope for improvement in terms of incorporation of time-evolution models by DMD. Existing DMDs either use the first-order ODE (exponential) model [2, 3, 12] or a neural network-based method to deal with the nonlinearity in time-evolution models [5]. Moreover, the time-evolution model can be extended within the linear model but with memory effects.

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

Although microscopic and fundamental physics are governed by integer-order ODEs (i.e., memoryless equations of motion), systems with strong coupling to an external system or *reservoir* behave differently. Consider that the internal state of the system is known and no microscopic information on the internal state of the reservoir is available. Upon applying a stimulus to the system, the system state changes, thus leading to a change in the reservoir state via the system-reservoir interactions. Subsequently, changes in the reservoir may also affect the system via the system-reservoir surface. Such an *indirect*

effect via an external system leads to a memory effect wherein the time evolution of the system appears to be affected by the *current* status and *history* of its time evolution [14, 15].

In the realm of materials science and engineering, memory effect is also known as *hysteresis*. It is also referred as the non-Markovian effects in some realms of science and engineering. Among the wide variety of phenomena that exhibits hysteresis, the $B-H$ response of the magnetic materials, friction [16], and piezoelectricity [17] are well-known examples in engineering. For instance, piezoelectric actuators are used in high-accuracy position controller in modern industrial machines (e.g., semiconductor processing equipment) and its behavior including the hysteresis is of great interest in control theory [17, 18].

As explained above, the *memory* of a system implies nonlocal behavior in the time domain, which mostly arises from the limitations of our observations. In the time-evolution equations explained in the previous section, memory effect corresponds to the nonlocality in the functional π. Assuming a linear functional, one can express π as follows:

$$\pi(\boldsymbol{x})(t) = \int \mathrm{d}t'\, g(t,t')\boldsymbol{x}(t'), \tag{15}$$

with $g(t,t')$ being the memory kernel. One example of the memory kernel is the power memory kernel [19]: $g(t,t') \propto (t-t')^{-\alpha}$ for $t' < t$. The power memory kernel corresponds to a memory weight which decays in time according to the power law, and it is shown that the power memory kernel leads to an equation of motion described by fractional differential equations (e.g., [20]). Another important example of memory kernel is the exponential memory kernel [21].

H. Study Aims

In this study, we proposed a DMD with *memory* (DMDm) method, whose time-evolution models are described by a wider class of equations that enable the description of the system with memory effects. As the basis of our discussion, we used the definition of the exact DMD given by Tu *et al.* (Definition 1 in [2]). Let us assume a linear functional relation, $y = \pi(x)$ for two time-series datasets, $x : t \mapsto x(t)$ and $y : t \mapsto y(t)$. As we have seen in section I F, if the value $y(t) = \pi(x)(t)$ is *not* affected by the values $x(t')$ for $t' > t$, then the linear map π is causal, because the operation is performed without knowing the *future* values of x.

Moreover, the construction of an eigenfunction of a causal linear operator can also be formulated as an IVP. This is also true in discrete-time-series data and in the discretized representation of the linear operator D_π, as shown in the next section. The discretized representation of the eigenfunction $z : t \mapsto z(t) \in \mathbf{R}$ corresponding to an eigenvalue λ is an array of numbers $\big[z(t_0), z(t_1), ..., z(t_{m-1})\big]$ that satisfy the following matrix equation:

$$\big[z(t_0), z(t_1), ..., z(t_{m-1})\big] D_\pi = \lambda \big[0, ..., 0, z(t_q), ..., z(t_{m-1})\big], \tag{16}$$

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

$$\boldsymbol{x}(t) = \sum_{j=0}^{r-1} \vec{\phi}_j F_{\pi,\lambda_j,z_{\mathrm{ini}}}(t). \tag{17}$$

In Eq. (17), the vector $\vec{\phi}_j \in \mathbf{R}^n$ is the DMD mode of the problem and the function $F_{\pi,\lambda,z_{\mathrm{ini}}}$ is the solution to the IVP $\pi(z)(t) = \lambda z(t)$, $\big[z(t_0), z(t_1), ..., z(t_{q-1})\big] = z_{\mathrm{ini}}$. Here, the eigenvector in the spatial direction $\vec{\phi}_j$ and eigenfunction in the time domain F_{π,λ_j} and z_{ini} share the common eigenvalue λ_j, corresponding to the factorization of the solution into temporal and spatial parts. In addition, the proposed framework contains the exact DMD, because

$$F_{\frac{\mathrm{d}}{\mathrm{d}t},\lambda,[1]} = \exp(\lambda t). \tag{18}$$

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 an example of this discussion, we use fractional calculus [14, 15, 22, 23] for time-domain transformation. The fractional derivatives of real-valued [15, 22, 23] and complex-valued orders [24] are useful in physics with memory effects using power law [14, 15]. The idea of introducing a fractional (non-integer)-order derivative has also attracted attention in control theory [25, 26]. One notable application of fractional calculus in control theory is the PI$^\lambda$D$^\mu$controller [26].

The α-th-order fractional integral of an integrable function f and real value $\alpha > 0$ are expressed as [22, 23].

$$(I^\alpha f)(t) = \frac{1}{\Gamma(\alpha)} \int_{-\infty}^{t} \mathrm{d}t'\, \frac{f(t')}{(t-t')^{1-\alpha}}, \tag{19}$$

where $\Gamma : \alpha \mapsto \Gamma(\alpha)$ is the gamma function. Note that the above definition corresponds to the case $g(t,t') = (t-t')^{\alpha-1}/\Gamma(\alpha)$, i.e., the power memory kernel.

The fractional integral satisfies the following properties for any integrable function f, g and the 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 of the order,

respectively. The third condition is the equivalence of I^α to the Riemann integral, for $\alpha = 1$. As the integral is the inverse of the derivative, the α-order derivative can be constructed for any $\alpha \in \mathbf{R}$ [22] for a smooth, integrable function $f : (-\infty, t_1] \to \mathbf{R}$ ($t_1 > -\infty$ is the upper bound of the domain of f).

Although the power-law in memory effects is just an approximation in some applications (e.g., [19]), the use of power memory kernel in terms of fractional derivative has two major advantages: 1) the set of fractional derivative operators forms a one-parameter group, 2) the implementation is relatively easy, because we can use the libraries for fractional derivatives.

In the following sections, we present the generic idea of DMDm and the detailed algorithm of *fractional* DMD or fracDMD. The fracDMD is a DMDm method wherein fractional differential equations determine the time dependency of each DMD mode instead of first-order differential equations. DMDm is a theoretical extension of DMD, enabling the analysis of multidimensional time-series data with memory effects in the DMD framework. Compared with existing DMD methods, our method has a wider degree of freedom in time-evolution models.

The remainder of this paper is organized as follows. In Section II, we investigate the properties of a causal linear function and its eigenfunctions. In Section III, we introduce the DMDm. In Sections IV and V, we introduce an arbitrary order DMD (fracDMD) as an example of the DMDm. In Sections VI and VII, we apply fracDMD to synthetic time-series data to demonstrate the validity of the proposed method. Further, Section VIII discusses the numerical results, and finally, Section IX concludes the study.

Throughout this study, we denote a $n \times m$ matrix as $V = [V_0, V_1, ..., V_{m-1}] = [V_{ij}]$ so that we can refer the column vectors V_i and elements V_{ij} with $0 \le i \le m-1$ and $0 \le j \le n-1$. We frequently use zero-padded matrices for a given matrix V e.g.,

$$\left[\mathbf{0}, ..., \mathbf{0}, V_q, .., V_{m-1}\right] \in \mathbf{R}^{n\times m}. \tag{20}$$

Hereafter, we assume that a zero-padded matrix is of the same shape as the original matrix, and thus one can assume that appropriate number of zero vectors are used.

II. CAUSAL LINEAR OPERATORS AND ITS PROPERTIES

In this section, we investigate the conditions for linear operator π, introduced in the previous section. Let us denote the set of maps $\{f : \mathcal{D} \to \mathcal{C}\}$ from set $\mathcal{D}$ to set $\mathcal{C}$ by $\hom(\mathcal{D}, \mathcal{C})$ or $\mathcal{C}^\mathcal{D}$.

For a particular set of time points $\mathcal{T} \subseteq \mathbf{R}$, we denote a set of one-dimensional (1D) 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; that is, $V : \mathcal{T} \to \mathbf{C}$.

A. Causal Linear Operator

Assume a set of time points $\mathcal{T}$ and its subset $\mathcal{S} \subseteq \mathcal{T}$. The original set $\mathcal{T}$ is either a closed interval in $\mathbf{R}$ or finite subset of $\mathbf{R}$. We define a *causal linear operator* $\pi : \mathbf{C}^\mathcal{T} \ni V \mapsto W \in \mathbf{C}^\mathcal{S}$ as a linear functional $\pi \in \hom(\mathbf{C}^\mathcal{T}, \mathbf{C}^\mathcal{S})$ between two time series that satisfy the *causality* condition. For a bounded set $\mathcal{T}$, we assume that $\min(\mathcal{T}) = 0$ and $\max(\mathcal{T}) = T > 0$ without loss of generality.

- Let $\widetilde{V}(-, t') : t \mapsto \widetilde{V}(t, t')$ is a function such that $\widetilde{V}(t, t') = V(t)\Theta(t' - t)$.
- Then, π is causal $\overset{\text{def}}{\Leftrightarrow} \pi(V)(t) = \pi(\widetilde{V}(-, t'))(t)$ 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 $t \ge 0$. We use a placeholder $-$ to distinguish the *function* $f(-)$ and *value* $f(x)$ for $f : x \mapsto f(x)$. Note that the more intuitive definition given in section I F coincides with the above definition.

B. Eigenfunction of the Causal Linear Operator

Hereafter, we assume that the causal linear operator π has a positive nullity $q \ge 1$, so that, by definition, we have to impose q initial condition(s). This means that the domain of the operator π is smaller than the set of time points we are interested in. Let the set of time points considered here be $\mathcal{T} = [0, T]$ for $T > 0$, and the domain of the operator π becomes $\mathcal{S} = (0, T]$. Note that the lack of $t = 0$ in $\mathcal{S}$ corresponds to the needs for the initial conditions for time-evolution equations of the form $\pi(\boldsymbol{x})(t) = f(\boldsymbol{x}(t), t)$.

For a particular causal linear operator $\pi \in \hom(\mathbf{C}^\mathcal{T}, \mathbf{C}^\mathcal{S})$ and the initial conditions, we can approximate the *eigenfunction* of π using an iterative method as follows. First, we introduce and fix a finite subset of $\mathcal{T}$ as shown below:

$$\mathsf{T} = \{t_i \in \mathcal{T} | i = 0, 1, 2, ..., m-1,\ t_i > t_j \text{ for } i > j,\ t_0 = 0,\ t_{m-1} = T\} \tag{21}$$

Furthermore, we introduce a (row) vector representation of time series V on T as $V = \left[V_0, V_1, ..., V_{m-1}\right]$, where $V_k = V(t_k) \in \mathbf{C}$ $(k = 0, 1, ..., m-1)$. Then, the matrix representation of π is introduced using the following equation:

$$VD_\pi = \pi(V), \tag{22}$$

where standard matrix multiplication is assumed on the left-hand side. Owing 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.

$$D_\pi = \begin{bmatrix} 0 & \cdots & 0 & * & \cdots & * & * \\ & \ddots & \vdots & * & \cdots & * & * \\ & & 0 & * & \cdots & * & * \\ & & & \pi_{qq} & & * & * \\ & & & & \ddots & \vdots & \vdots \\ & & & & & \pi_{m-2,m-2} & * \\ & \boldsymbol{O} & & & & 0 & \pi_{m-1,m-1} \end{bmatrix}. \tag{23}$$

Hereafter, we assume that the product of the lower $m-q$ diagonal elements is nonzero, that is, $\pi_{qq}\pi_{q+1,q+1}\cdots\pi_{m-1,m-1} \neq 0$. The integer q corresponds to 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 by the following matrix equations in association with the initial condition for the row vector V and matrix D_π:

$$VD_\pi = \lambda\big[\boldsymbol{0}, ..., \boldsymbol{0}, V_q, V_{q+1}, ..., V_{m-1}\big], \tag{24}$$

$$\big[V_0, V_1, ..., V_{q-1}\big]\Omega = \boldsymbol{\gamma}, \tag{25}$$

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 IVP Eq. (24) and Eq. (25) to be solvable, we must impose that $\mathrm{rank}(\Omega) = q$. Note that the second equation Eq. (25) corresponds to the initial conditions, specifying the first q components of vector V, and 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 initial conditions is to assign the first q values to the time series V:

$$[V_0, V_1, ..., V_{q-1}] = [V_{\mathrm{ini},0}, V_{\mathrm{ini},1}, ..., V_{\mathrm{ini},q-1}], \tag{26}$$

corresponding to the cases $\Omega = \hat{1}_q$ and $\boldsymbol{\gamma} = [V_{\mathrm{ini},0}, V_{\mathrm{ini},1}, ..., V_{\mathrm{ini},q-1}]$ in Eq. (25).

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

$$V_k = \sum_{j=0}^{p-1} \big(\Omega^+\big)_{jk}\gamma_j, \tag{27}$$

and for $k = q, q+1, ..., m-1$, solving Eq. (24) for V_k in terms of $V_0, ..., V_{k-1}$ yields the iterative equation:

$$V_k = \left[\frac{\big[V_0, ..., V_{k-1}, \boldsymbol{0}, ..., \boldsymbol{0}\big]D_\pi}{\lambda - \pi_{kk}}\right]_k. \tag{28}$$

The procedure to obtain the approximated eigenmode of the causal linear operator π from the first q values $V_0, V_1, ..., V_{q-1}$ is summarized in Algorithm 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 λ, initial values $V_{\mathrm{ini}} = \big[V_0, V_1 ... V_{q-1}\big] \in \mathbf{C}^q$, matrix representation D_π corresponding to the discrete set of time points 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**
    $V_k \leftarrow \left[\frac{\big[V_0, ..., V_{k-1}, \boldsymbol{0}, ..., \boldsymbol{0}\big]D_\pi}{\lambda - \pi_{kk}}\right]_k$
    $V \leftarrow \big[V_0, V_1, ..., V_k, 0, ..., 0\big] \in \mathbf{C}^m$
  end for
  $\boldsymbol{\phi}_{\pi,\lambda,V_{\mathrm{ini}}} \leftarrow V$
  return $\boldsymbol{\phi}_{\pi,\lambda,V_{\mathrm{ini}}}$

By using a sequence of time points T with the maximum time interval $\Delta t_{\max} \to +0$, Algorithm 1 can approximate a smooth function. To ensure that Algorithm 1 functions properly, we must identify a matrix D_π whose first q columns are zero vector, whereas the other column vectors are linearly independent.

III. DYNAMIC MODE DECOMPOSITION WITH MEMORY

Using the causal linear operator and its eigenfunctions introduced in the previous section, we postulated the concept of DMDm, which is a new method that includes memory effects in the DMD framework. It employs a causal linear operator instead of the difference operator used in existing DMD. This enabled us to handle the effects of past data in the time-evolution model without losing the advantages of the DMD framework. Throughout this section, we define the intervals $\mathcal{T}, \mathcal{S} \subset \mathbf{R}$ as $\mathcal{T} = [0, T]$ and $\mathcal{S} = (0, T] \subset \mathcal{T}$ for $T > 0$.

A. Model Definition

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

$$\pi(\boldsymbol{x})(t) = A\boldsymbol{x}(t) \quad (t \in \mathcal{S}), \tag{29}$$

where $A \in \mathbf{R}^{n\times n}$ denotes the constant matrix. By restricting t to the (finite) set of time points $\mathsf{T} = \{t_i \in \mathcal{T} | i = 0, 1, ..., m-1,\ t_0 = 0,\ t_{m-1} = T,\ t_i > t_j \text{ for } i > j\} \subset \mathcal{T}$, the continuous time-evolution model can be discretized to obtain the matrix representation

$$XD_\pi = A\big[\boldsymbol{0}, ..., \boldsymbol{0}, \boldsymbol{x}_q, ..., \boldsymbol{x}_{m-1}\big], \tag{30}$$

where matrix A and n-dimensional time-series data X are used instead of scalar λ and 1D time-series V in Eq. (24). The nullity q depends on the nature of operator π. Considering the use of (full) SVD $X = U\Sigma V^*$ with the unitary (square) matrices U and V, we can transform Eq. (30) as follows:

$$U^*XD_\pi = U^*AU\big(U^*\big[\boldsymbol{0}, ..., \boldsymbol{0}, \boldsymbol{x}_q, ..., \boldsymbol{x}_{m-1}\big]\big). \tag{31}$$

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

$$\Xi D_\pi = \mathrm{diag}(\lambda_0, ..., \lambda_{n-1})\big[\mathbf{0}, ..., \mathbf{0}, \boldsymbol{\xi}_q, ..., \boldsymbol{\xi}_{m-1}\big]. \quad (32)$$

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

$$\begin{aligned}&\big[\boldsymbol{x}(t_0), \boldsymbol{x}(t_1), ..., \boldsymbol{x}(t_{m-1})\big] \\ &= U\mathcal{U}\begin{bmatrix} \boldsymbol{\phi}^\top_{\pi,\lambda_0,\xi_{0,\mathrm{ini}}} \\ \vdots \\ \boldsymbol{\phi}^\top_{\pi,\lambda_{n-1},\xi_{n-1,\mathrm{ini}}}\end{bmatrix} \\ &= \sum_{i=0}^{n-1}(U\mathcal{U}\mathbf{e}_i)\boldsymbol{\phi}^\top_{\pi,\lambda_i,\xi_{i,\mathrm{ini}}}, \end{aligned} \quad (33)$$

where the column vector $\boldsymbol{\phi}_{\pi,\lambda_0,\xi_{i,\mathrm{ini}}}$ is defined in Algorithm 1 and $\mathbf{e}_i = \big[\delta_{0i}, \delta_{1i}, ..., \delta_{n-1,i}\big]^\top$ with the Kronecker delta δ_{ij} ($\delta_{ii} = 1$ and $\delta_{ij} = 1$ for $i \neq j$), and $\xi_{i,\mathrm{ini}} = \big[\xi_{i,0}, \xi_{i,1}, ..., \xi_{i,q-1}\big]$ is a vector comprising the values of $\xi_i = \mathbf{e}_i \cdot \boldsymbol{\xi}$ at the first q time points. The expression above is similar to that of Eq. (17).

B. Model Fitting based on DMD Scheme

Our goal is to determine an appropriate matrix A that can be used to explain the data X on a discrete set of time points T. We assume that model Eq. (30) holds for the observed data X. The best-fit parameter A that achieves least squares for $\pi(x)(t)$ is estimated as follows:

$$\begin{aligned} A &= \arg\min_{A'} \big\| XD_\pi - A'\big[\mathbf{0}, ..., \mathbf{0}, \boldsymbol{x}_q, ..., \boldsymbol{x}_{m-1}\big]\big\| \\ &= XD_\pi\big[\mathbf{0}, ..., \mathbf{0}, \boldsymbol{x}_q, ..., \boldsymbol{x}_{m-1}\big]^+. \end{aligned} \quad (34)$$

Similar to Eq. (3), we can compute A using SVD $\big[\mathbf{0}, ..., \mathbf{0}, \boldsymbol{x}_q, ..., \boldsymbol{x}_{m-1}\big] = U\Sigma V^*$ to obtain an explicit expression for DMD with memory:

$$A = XD_\pi V\Sigma^+U^*. \quad (35)$$

The low-rank approximation can also be performed in a manner similar to that in Eq. (4).

C. DMDc with Memory

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

$$\pi(\boldsymbol{x})(t) = A\boldsymbol{x}(t) + B\boldsymbol{u}(t) \quad (t \in \mathcal{S}), \quad (36)$$

which corresponds to the matrix representation.

$$XD_\pi = A\big[\mathbf{0}, ..., \mathbf{0}, \boldsymbol{x}_q, ..., \boldsymbol{x}_{m-1}\big] + B\big[\mathbf{0}, ..., \mathbf{0}, \boldsymbol{u}_q, ..., \boldsymbol{u}_{m-1}\big], \quad (37)$$

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 provides direct calculation of the optimal coefficients, similar to Eq. (10), as

$$\big[A\ B\big] = XD_\pi\begin{bmatrix}\big[\mathbf{0}, ..., \mathbf{0}, \boldsymbol{x}_q, ..., \boldsymbol{x}_{m-1}\big] \\ \big[\mathbf{0}, ..., \mathbf{0}, \boldsymbol{u}_q, ..., \boldsymbol{u}_{m-1}\big]\end{bmatrix}^+. \quad (38)$$

Algorithm 1 is readily applicable for the multidimensional case, and the time-evolution of the model Eq. (37) for a given initial condition $X\Omega = \Gamma$, external force $\Upsilon = [\boldsymbol{u}_0, \boldsymbol{u}_1, ..., \boldsymbol{u}_{m-1}]$, and coefficient matrices A, B is obtained by the following successive calculations: for $k = 0, 1, ..., q-1$,

$$X_k = \sum_{j=0}^{p-1}\big(\Omega^+\big)_{jk}\Gamma_j, \quad (39)$$

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

$$X_k = \left[\frac{(A\big[\boldsymbol{x}_0, ..., \boldsymbol{x}_{k-1}, \mathbf{0}, ..., \mathbf{0}\big] + B\big[\boldsymbol{u}_0, ..., \boldsymbol{u}_{k-1}, \mathbf{0}, ..., \mathbf{0}\big])D_\pi}{A - \pi_{kk}\hat{1}_n}\right]_k. \quad (40)$$

where the fractions of matrices $G \in \mathbf{C}^{n\times m}$ and $H \in \mathbf{C}^{n\times n}$ are defined by $G/H = H^{-1}G$. This is a natural extension of DMDc.

IV. FRACTIONAL DMD

The order of the differential operator can be extended to any real-valued number [22, 23]. The term for this generalized differentiation, the *fractional derivative*, is actually misleading. This is because we can also specify an irrational number $\alpha \in \mathbf{R}\backslash\mathbf{Q}$ as the order of differentiation. One of the definitions of the α-th order differential of a smooth integrable function $f : \mathbf{R} \to \mathbf{R}$ is the Caputo derivative. Specifically, for $t > 0$,

$$\frac{\mathrm{d}^\alpha f}{\mathrm{d}t^\alpha} = \begin{cases}\frac{1}{\Gamma(\nu)}\int_0^t \mathrm{d}t'\,\frac{f^{(n_\alpha^+)}(t')}{(t-t')^{1-\nu}} & (\alpha \notin \mathbf{Z}_{\geq 0}) \\ f^{(\alpha)}(t) & (\alpha \in \mathbf{Z}_{\geq 0})\end{cases} \quad (41)$$

where $n_\alpha^+ = \max(0, \lceil\alpha\rceil)$, $\nu = n_\alpha^+ - \alpha$, $f^{(\ell)}$ 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 $y - x \geq 0$. The exception for $\alpha \in \mathbf{Z}_{\geq 0}$ (the second line in Eq. (41)) is not necessary because the two cases in Eq. (41) coincide at $\alpha \to \alpha_0 \in \mathbf{Z}_{\geq 0}$, both of which yield $\frac{\mathrm{d}^\ell f}{\mathrm{d}t^\ell} = f^{(\ell)}$ for $\ell \in \mathbf{Z}_{\geq 0}$. Moreover, the right-hand side of Eq. (41) with $\alpha \notin \mathbf{Z}_{\geq 0}$ coincides with the definition of the fractional *integral* in Eq. (19) with the replacements $\alpha \mapsto \nu$ and $f \mapsto f^{(\ell)}$ and $0 < \nu < 1$. In this section, we consider real-valued functions for simplicity.

We can construct the corresponding *eigenmode* for the α-th order differential operator with an initial condition. To demonstrate this, we denote the 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 $D_m^{(\ell)}$ be an N-th order approximation of the ℓ-th order integer-order derivative: For a smooth function $f : \mathbf{C} \to \mathbf{C}$ and finite set of time points $\mathsf{T} = \{t_i \in \mathbf{R} | i = 0, 1, ..., m-1, t_i > t_j \text{ for } i > j\}$.

$$D_m^{(\ell)} \begin{bmatrix} f(t_0) \\ \vdots \\ f(t_{\ell-1}) \\ f(t_\ell) \\ \vdots \\ f(t_{m-1}) \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ f^{(\ell)}(t_\ell) + \mathcal{O}\big((\Delta t_{\max})^N\big) \\ \vdots \\ f^{(\ell)}(t_{m-1}) + \mathcal{O}\big((\Delta t_{\max})^N\big) \end{bmatrix}, \quad (42)$$

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. Here, $\mathrm{rank}(D_m^{(\ell)}) = m - \ell$. The explicit expression of $\mathcal{D}_m^{(\alpha)}$ is now constructed using the Caputo fractional differential [22] as follows:

$$\mathcal{D}_m^{(\alpha)} = D_m^{(n_\alpha^+)} \Big[\boldsymbol{w}_0^{(n_\alpha^+ - \alpha)}, \boldsymbol{w}_1^{(n_\alpha^+ - \alpha)}, ..., \boldsymbol{w}_{m-1}^{(n_\alpha^+ - \alpha)} \Big]. \quad (43)$$

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

$$(\boldsymbol{w}_k^{(\nu)})_i = \begin{cases} \frac{1}{\Gamma(\nu+1)}((t_{k+1} - t_i)^\nu - (t_{k+1} - t_{i+1})^\nu) & (i \leq k) \\ 0 & (i > k) \end{cases}. \quad (44)$$

where $\Delta t_i = t_{i+1} - t_i$. The approximation error Eq. (44) is $\mathcal{O}(T \Delta t_{\max})$, where $T = t_{m-1} - t_0$ is the total time. The matrix representation for the fractional derivative is then decomposed into the ℓ-th order (integer-order) differential and the $(-\nu)$-th order fractional derivative (i.e., the fractional integral of order $0 < \nu < 1$). Clearly, the discretized representation $\mathcal{D}^{(\alpha)}$ satisfies linearity and causality, thereby ensuring the existence of eigenmodes constructed by Algorithm 1. More elaborate implementations with higher-order schemes may be used in an actual numerical analysis. In the continuous limit, the eigenfunction of the fractional derivative operator is expressed by the Mittag-Leffler function [23, 27, 28].

Because $\mathrm{rank}(D_m^{(\ell)}) = m - \ell$ and $\{\boldsymbol{w}_k^{(\nu)} | k = 0, 1, ..., m-1\}$ is a set of m linearly independent vectors for $\nu \neq 1$, we 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, coupled with the initial condition $\big[\boldsymbol{x}_0, .., \boldsymbol{x}_{n_\alpha^+ - 1}\big]\Omega = \Gamma$ for $\Omega \in \mathbf{C}^{n_\alpha^+ \times p}$ and $\Gamma \in \mathbf{C}^{n\times p}$ is expressed as follows:

$$X\mathcal{D}^{(\alpha)} = A\big[\boldsymbol{0}, ..., \boldsymbol{0}, \boldsymbol{x}_{n_\alpha^+}, ..., \boldsymbol{x}_{m-1}\big]; \quad \big[\boldsymbol{x}_0, .., \boldsymbol{x}_{n_\alpha^+ - 1}\big]\Omega = \Gamma. \quad (45)$$

For the given time-series data $X \in \mathbf{R}^{n\times m}$ arranged in matrix form and a fixed order of derivative α, the best-fit matrix A for the model equation Eq. (45) is obtained as follows:

$$A = X\mathcal{D}^{(\alpha)}\big[\boldsymbol{0}, ..., \boldsymbol{0}, \boldsymbol{x}_{n_\alpha^+}, ..., \boldsymbol{x}_{m-1}\big]^+. \quad (46)$$

The expression above is analogous to Eq. (3) for a 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 $\hat{\Sigma}, \hat{\Sigma}_r$ yield a low-rank representation of the dynamics for $\boldsymbol{\xi} = \hat{U}_r^*\boldsymbol{x} \in \mathbf{R}^r$.

$$\frac{\mathrm{d}^\alpha \boldsymbol{\xi}}{\mathrm{d}t^\alpha} = \Lambda\boldsymbol{\xi}, \quad \Lambda = \hat{U}_r^* X\mathcal{D}^{(\alpha)}\hat{V}_r\hat{\Sigma}_r^{-1}. \quad (47)$$

If the order α is unknown, matrix A_* and the optimal fractional order α_* is estimated using Algorithm 2.

Algorithm 2 Grid search for order α

Require: Time-series data in matrix form X, candidates $\boldsymbol{\alpha} = \{\alpha_0, \alpha_1, ...\}$
  for $\alpha \in \boldsymbol{\alpha}$ **do**
    $A \leftarrow X\mathcal{D}^{(\alpha)}\big[\boldsymbol{0}, ..., \boldsymbol{0}, \boldsymbol{x}_{n_\alpha^+}, ..., \boldsymbol{x}_{m-1}\big]^+$
    $L(\alpha) \leftarrow \Big\| X\mathcal{D}^{(\alpha)} - A\big[\boldsymbol{0}, ..., \boldsymbol{0}, \boldsymbol{x}_{n_\alpha^+}, ..., \boldsymbol{x}_{m-1}\big]\Big\|$
  end for
  $\alpha_* \leftarrow \arg\min_\alpha (L(\alpha))$
  $A_* \leftarrow X\mathcal{D}^{(\alpha_*)}\big[\boldsymbol{0}, ..., \boldsymbol{0}, \boldsymbol{x}_{n_\alpha^+}, ..., \boldsymbol{x}_{m-1}\big]$

The above algorithm is to find the optimal α_* that minimizes the reconstruction error $L(\alpha) = \big\|X\mathcal{D}^{(\alpha)} - A\big[\boldsymbol{0}, ..., \boldsymbol{0}, \boldsymbol{x}_{n_\alpha^+}, ..., \boldsymbol{x}_{m-1}\big]\big\|$ with A being the optimal coefficient matrix for a given α: $A = X\mathcal{D}^{(\alpha)}\big[\boldsymbol{0}, ..., \boldsymbol{0}, \boldsymbol{x}_{n_\alpha^+}, ..., \boldsymbol{x}_{m-1}\big]^+$. Using this algorithm, one can find the optimal order and coefficient without using any *a priori* information.

V. FRACTIONAL DMD WITH CONTROL

The fracDMD proposed in the previous section is only valid for autonomous dynamical systems. Analogous to DMDc [12], we can extend our method to non-autonomous systems with input terms. We assume that $\mathcal{T} = [0, T]$ and $\mathcal{S} = (0, T] \subset \mathcal{T}$. The Caputo derivative is a causal linear operator, $\frac{\mathrm{d}^\alpha}{\mathrm{d}t^\alpha} : \mathcal{T} \to \mathcal{S}$. Consider the following dynamical system:

$$\frac{\mathrm{d}^\alpha \boldsymbol{x}}{\mathrm{d}t^\alpha} = A\boldsymbol{x} + B\boldsymbol{u}(t) \quad (t \in \mathcal{S}), \quad (48)$$

with the state vector $\boldsymbol{x} \in \mathbf{R}^n$ and an external force (exogenous input) vector $\boldsymbol{u}(t) \in \mathbf{R}^\ell$. Matrices $X = [\boldsymbol{x}_0, \boldsymbol{x}_1, ..., \boldsymbol{x}_{m-1}]$ and $\Upsilon = [\boldsymbol{u}_0, \boldsymbol{u}_1, ..., \boldsymbol{u}_{m-1}]$ were constructed. Similar to Eq. (10), we obtained the following estimation for the coefficients.

$$\big[A\ B\big] \simeq \big[\bar{A}\ \bar{B}\big] = X\mathcal{D}^{(\alpha)}\begin{bmatrix} \big[\boldsymbol{0}, ..., \boldsymbol{0}, \boldsymbol{x}_{n_\alpha^+}, ..., \boldsymbol{x}_{m-1}\big] \\ \big[\boldsymbol{0}, ..., \boldsymbol{0}, \boldsymbol{u}_{n_\alpha^+}, ..., \boldsymbol{u}_{m-1}\big] \end{bmatrix}^+. \quad (49)$$

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 $p > r$ can be used to obtain an approximate low-rank representation of the dynamics for $\boldsymbol{\xi} = \hat{U}^*\boldsymbol{x} \in \mathbf{R}^r$, as follows:

$$\frac{\mathrm{d}^\alpha \boldsymbol{\xi}}{\mathrm{d}t^\alpha} = \Lambda\boldsymbol{\xi} + \Gamma\boldsymbol{u}, \quad [\Lambda \ \ \Gamma] \simeq \hat{U}_r^* X\mathcal{D}^{(\alpha)}\tilde{V}_p\tilde{\Sigma}_p^{-1}\tilde{U}_p^*\begin{bmatrix}\hat{U}_r^* & 0\\ 0 & \hat{1}_{p-r}\end{bmatrix}. \tag{50}$$

Thus, the time-series data of the input $\boldsymbol{u}(t)$ and output $\boldsymbol{x}(t)$ can be analyzed for a system in the fracDMD framework, as well as for autonomous systems. The fracDMD algorithm is described explicitly in Algorithm 3. Function $\mathtt{SVD}(\bullet, r)$ denotes the SVD of a matrix with rank r.

Algorithm 3 FracDMD with control

Require: Input data in matrix form Υ, observation data in matrix form X, ranks $p > r > 0$, and order of the fractional differential equation α

$\hat{U}_r, \hat{\Sigma}_r, \hat{V}_r \leftarrow \mathtt{SVD}(\big[\boldsymbol{0}, ..., \boldsymbol{0}, \boldsymbol{x}_{n_\alpha^+}, ..., \boldsymbol{x}_{m-1}\big], r)$

$\tilde{U}_p, \tilde{\Sigma}_p, \tilde{V}_p \leftarrow \mathtt{SVD}(\big[\big[\boldsymbol{0}, ..., \boldsymbol{0}, \boldsymbol{x}_{n_\alpha^+}, ..., \boldsymbol{x}_{m-1}\big]; \big[\boldsymbol{0}, ..., \boldsymbol{0}, \boldsymbol{u}_{n_\alpha^+}, ..., \boldsymbol{u}_{m-1}\big]\big], p)$

$X' \leftarrow X\mathcal{D}^{(\alpha)}$

$[\Lambda \ \ \Gamma] \leftarrow \hat{U}_r^* X' \tilde{V}_p\tilde{\Sigma}_p^{-1}\tilde{U}_p^*\begin{bmatrix}\hat{U}_r^* & 0\\ 0 & \hat{1}_{p-r}\end{bmatrix}$

VI. QUANTITATIVE MODEL EVALUATIONS

The fracDMD and existing DMD minimize the Frobenius norm of the matrix $\|X' - AX - B\Upsilon\|$, where X' is the fractional derivative $X\mathcal{D}^{(\alpha)}$ for 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 the squared reconstruction error in α-th order time derivative, with $AX + B\Upsilon$ regarded as the reconstruction by the model.

Another method of evaluating 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 the 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

$$\mathcal{L}^{(\alpha)}_{\text{SSE}}(A, B; X) = \sum_{i=0}^{m-1} |\boldsymbol{x}(t_i) - \boldsymbol{x}_i|^2, \tag{51}$$

where $\boldsymbol{x}(t_i)$ is the solution of the time-evolution equation Eq. (48) with the coefficients A and B and the initial condition, whose explicit expression is shown in Eq. (33).

VII. NUMERICAL EXPERIMENTS

We performed numerical experiments to demonstrate the utility of the fracDMD against synthetic data. We numerically generated a solution for the fractional oscillator [29] analyzed by Svenkeson *et al.* [14] in the 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 demonstrate the utility of fractional-order calculus in analyzing memory effects. We extended their method to include multidimensional noisy fractional oscillators. In addition, we used an observation matrix $R \neq \hat{1}$ such that the mode reconstruction became highly nontrivial.

A. Numerical Setup

A 1D fractional linear oscillator is expressed by the following equation of motion [14]:

$$\frac{\mathrm{d}^\nu \boldsymbol{y}_a}{\mathrm{d}t^\nu} = Q_a\boldsymbol{y}_a; \quad \boldsymbol{y}_a = \begin{bmatrix}x_a\\ v_a\end{bmatrix}, \quad Q_a = \begin{bmatrix}0 & 1\\ -\omega_a^2 & 0\end{bmatrix} \quad (a = 0, 1, ..., k-1), \tag{52}$$

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

$$\frac{\mathrm{d}^\nu \boldsymbol{x}}{\mathrm{d}t^\nu} = \Phi\boldsymbol{x}; \quad \boldsymbol{x} = \begin{bmatrix}\boldsymbol{y}_0\\ \boldsymbol{y}_1\\ \vdots\\ \boldsymbol{y}_{k-1}\end{bmatrix} \in \mathbf{R}^{2k}, \tag{53}$$

$$\Phi = \begin{bmatrix}Q_0 & & & O\\ & Q_1 & & \\ & & \ddots & \\ O & & & Q_{k-1}\end{bmatrix} \in \mathbf{R}^{2k\times 2k}, \tag{54}$$

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

$$\boldsymbol{z}(t) = R\boldsymbol{x}(t) + \boldsymbol{\epsilon}(t), \quad \epsilon_i(t) \overset{\text{iid}}{\sim} \mathcal{N}(0, \sigma^2) \quad (i = 0, 1, ..., n-1), \tag{55}$$

where $R \in \mathbf{R}^{n\times 2k}$ is the constant observation matrix, and each element of $\boldsymbol{\epsilon}(t) = \big[\epsilon_0(t), \epsilon_1(t), ..., \epsilon_{n-1}(t)\big] \in \mathbf{R}^n$ is the independent and identically distributed (iid) Gaussian noise, and $\boldsymbol{z} \in \mathbf{R}^n$ represents the observed signal. This numerical setup is useful for describing a situation wherein the oscillators do not interact with each other; however, the resulting signal is the superposition of the oscillators.

Hereafter, we assume $n \geq 2k$. The observation matrix R comprises 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:

$$R = \big[v_{s(0)}, v_{s(1)}, ..., v_{s(2k-1)}\big] \in \mathbf{R}^{n\times 2k}. \tag{56}$$

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

B. Numerical Tests

Numerical tests were performed to determine the frequency ω_a $(a = 0, 1, ..., k-1)$ in Eq. (52) and Eq. (53). The eigenvalues of 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 using fracDMD as λ_i $(i = 0, 1, ..., 2k-1)$. It is assumed that $\omega_0 > \omega_1 > ... > \omega_{k-1} > 0$ and $\mathrm{Im}\,\lambda_0 > \mathrm{Im}\,\lambda_1 > ... > \mathrm{Im}\,\lambda_{k-1} > \mathrm{Im}\,\lambda_{2k-1} > \mathrm{Im}\,\lambda_{2k-2} > ... > \mathrm{Im}\,\lambda_k$.

The error L in the frequency estimation is expressed as follows:

$$L = \sum_{i=0}^{k-1} |\lambda_i - \omega_i|^2 + \sum_{i=0}^{k-1} |\lambda_{i+k} + \omega_i|^2. \tag{57}$$

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

We show the frequency reconstruction error L for various values of the noise standard deviation σ and the dimensions of the observation vectors n in Fig. 1 and Fig. 2 below. We assume that rank $2k$ and order ν of the system equation Eq. (53) are known. We used second-order numerical discretization of the Caputo fractional differential instead of the first-order scheme shown in Eq. (43) and Eq. (44). We also modified Eq. (46) to use the fractional *integral* instead of the fractional differential with a positive order α to achieve better numerical convergence. For details on the numerical implementations, see Appendix B.

We can construct the observed data in two ways: one is to give the analytical solution using Mittag-Leffler functions with the given coefficient matrices, and the other is to give the result of the numerical time-evolution using Algorithm 1. The numerical solution of the time-evolution equation has discretization errors, and thus does not coincide with the analytical solutions.

The estimated coefficients by fracDMD are optimized for the numerical time-evolution scheme with finite time-step, and thus we have a greater reconstruction error for the analytically obtained observation data, as shown in Fig. 2 and Fig. 1. In other words, a reconstruction error of the analytically generated (continuous) time-series data includes both discretization error L_{disc} and modeling error L_{model}, whereas we can suppress L_{disc} in a case with numerically generated observation data.

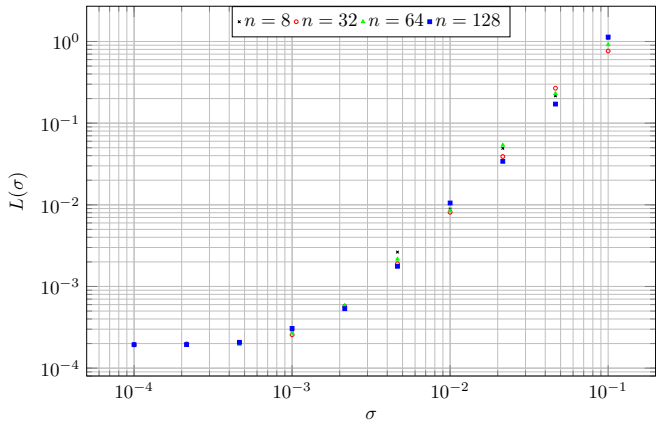

FIG. 1. Frequency reconstruction error L (Eq. (57)) for the analytically generated observation data. We plot the error L as a function of the noise standard deviation σ for various values of observation size n. The numerical setups are $k = 4$, $\omega_i = i + 1$ $(i = 0, 1, 2, 3)$, and the order of the system equation Eq. (53) 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 such that the initial (pseudo) energy of each oscillator becomes unity. We discretize the time interval $[0, 5]$ to 100 time points, and we use all time points in fracDMD. The observed data is calculated using the Mittag-Leffler functions with the coefficients specified above.

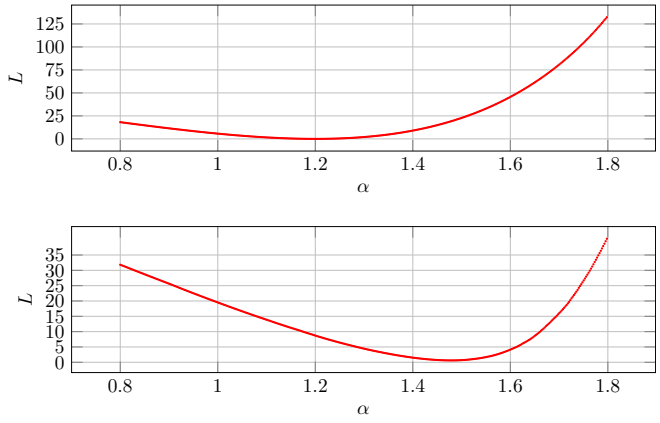

FIG. 3. Frequency reconstruction error L as a function of the order α used in the fracDMD. Upper panel: $\nu = 1.2$. Lower panel: $\nu = 1.5$. In both tests, $\sigma = 10^{-2}$ and $n = 8$. The other numerical setups are the same as in Fig. 1. The SVD rank r is set to the actual system size $2k$. The achieved numerical minima are at $\alpha = 1.200$ for (a) and $\alpha = 1.478$ for (b).

Fig. 3 shows the function $L(\alpha)$ with its minima located approximately at $\alpha = \nu$, thus indicating that an *erroneous* value of α leads to a larger reconstruction error value.

We also show the results of the frequency reconstruc-

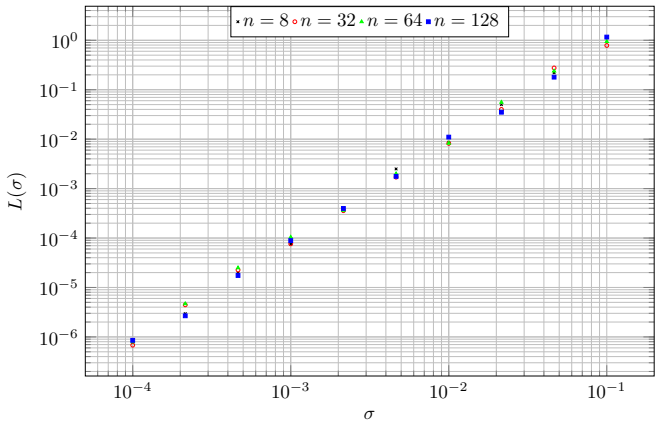

FIG. 2. Frequency reconstruction error L (Eq. (57)) for the numerically generated observation data. We plot the error L as a function of the noise standard deviation σ for various values of observation size n. The numerical setups are the same as in Fig. 1. The observed data is calculated by Algorithm 1 with the coefficients specified above.

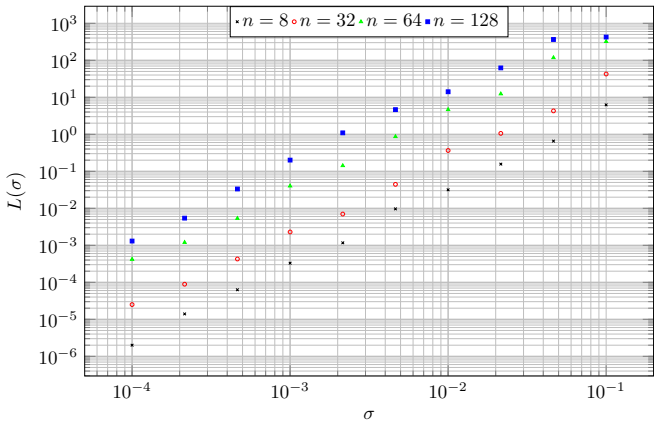

FIG. 4. Frequency reconstruction error L as a function of the noise standard deviation σ for various values of observation size n for the system of fractional oscillators with external force. The external force is the step-wise function shown in Eq. (58). The other numerical set-ups are the same as in Fig. 1. The observed data is calculated by Eq. (40) with the coefficients specified above.

tion for the case with nonzero external forces $\boldsymbol{u} \in \mathbf{R}^2$. We used three different profiles of $\boldsymbol{u}$, as shown below.

$$\boldsymbol{u}(t) = \begin{cases} \begin{bmatrix} 0.4\Theta(t-2.5) \\ 0.2\Theta(t-1.25)\Theta(3.75-t) \end{bmatrix} & \text{(step-wise)} \\ \begin{bmatrix} 1 \\ 1 \end{bmatrix} \delta(t) & \text{(impulse)} \\ \begin{bmatrix} 1 \\ 1 \end{bmatrix} & \text{(constant)} \end{cases} . \tag{58}$$

We set the coefficient B as follows:

$$B = \frac{1}{4} \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} . \tag{59}$$

The results are shown in Fig. 4 and Fig. 5 below.

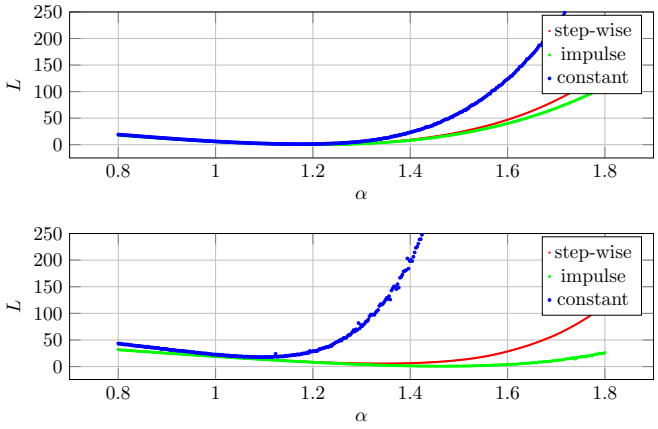

FIG. 5. Frequency reconstruction error L as a function of the order α used in the fracDMD, with external forces defined in Eq. (58). Upper panel: $\nu = 1.2$. Lower panel: $\nu = 1.5$. In both tests, $\sigma = 10^{-2}$ and $n = 8$. The other numerical setups are the same as in Fig. 4. The SVD rank r is set to the actual system size $2k$.

In Fig. 4, we can see the similar tendency to the case without external forces (Fig. 2) for the reconstruction error $L = L(\sigma)$, whereas the greater errors are seen in the case with external forces. Fig. 5 shows that the achieved minima is hugely dependent on the input profile $\boldsymbol{u} = \boldsymbol{u}(t)$, and the estimation becomes worse in the case $\nu = 1.5$. The constant input tends to have greater errors, while step-wise and impulse inputs have better reconstructions.

C. Numerical Tests with Partial Observations

We next perform a more realistic numerical demonstration with a larger system of fractional oscillator. We assume that the observation matrix R is now a $n \times 2k$ matrix with $n < 2k$, so the observation is now partial. We also assume that $k = r_d d$ for $d, r_d \in \mathbf{Z}_{>0}$. We set the oscillator frequencies as follows: using the division remainder function $\mathrm{rem}(-,-)$ such that $\mathrm{rem}(a,b)$ is the remainder of a divided by b for $a, b \in \mathbf{Z}_{>0}$,

$$\omega_i = \mathrm{rem}(i, d) + 1 \quad (i = 0, 1, ..., k-1). \tag{60}$$

Note that the each frequency repeats r_d times in Φ. This is to mimic a realistic physical situation in which several oscillation modes share common frequencies. We use no external force in this subsection.

Let us construct the observation matrix R as follows: using a random orthogonal basis of $2k$-dimensional vector space $\{w_0, w_1, ..., w_{2k}\}$ and a permutation $s \in \mathfrak{S}_{2k}$,

$$R = [w_{s(0)}, w_{s(1)}, ..., w_{s(n)}]^\top \in \mathbf{R}^{n \times 2k}. \tag{61}$$

The observed data is constructed using Eq. (55). For each value of n, we performed 30 times of numerical experiments with different realization of observation matrix R. The reconstruction error (RMSE) against the observed data is calculated for each numerical experiment and we plot the mean RMSE against the observation dimension n in Fig. 6.

We can see that the reconstruction is relatively good for $n \gtrsim 2d$. Note that the number of independent time-evolution modes is $2d$, because we have d unique frequencies in Φ and each frequency corresponds to a two-dimensional subspace in the state space. In this sense, the intrinsic dimension of the system is $2d$.

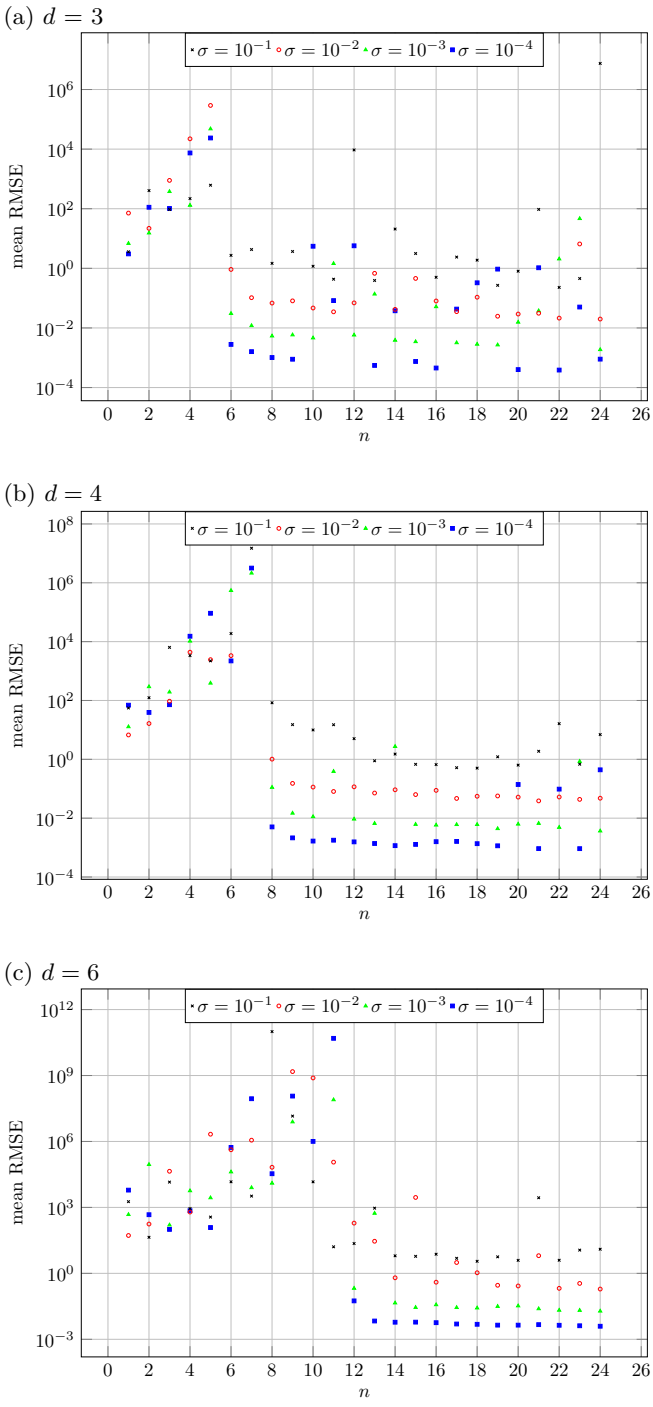

FIG. 6. Reconstruction error (mean RMSE) as a function of the number of observations n used in the fracDMD. The ground truth is the observation data. The parameters are as follows: (a) $d = 3, r_d = 4$, (b) $d = 4, r_d = 3$, (c) $d = 6, r_d = 2$. The mean is taken over 30 iterations with different realization of the observation matrix R. The SVD rank r is set to the actual observation size n.

Examples of the reconstructed time-series data for $n = 3$ and $n = 4$ with $\sigma = 10^{-4}$ are shown in Fig. 7 and in Fig. 8, respectively. We can see that the reconstructions are successful for $n \geq 2d$ cases, while the reconstructions

significantly deviate from the observed data for $n < 2d$.

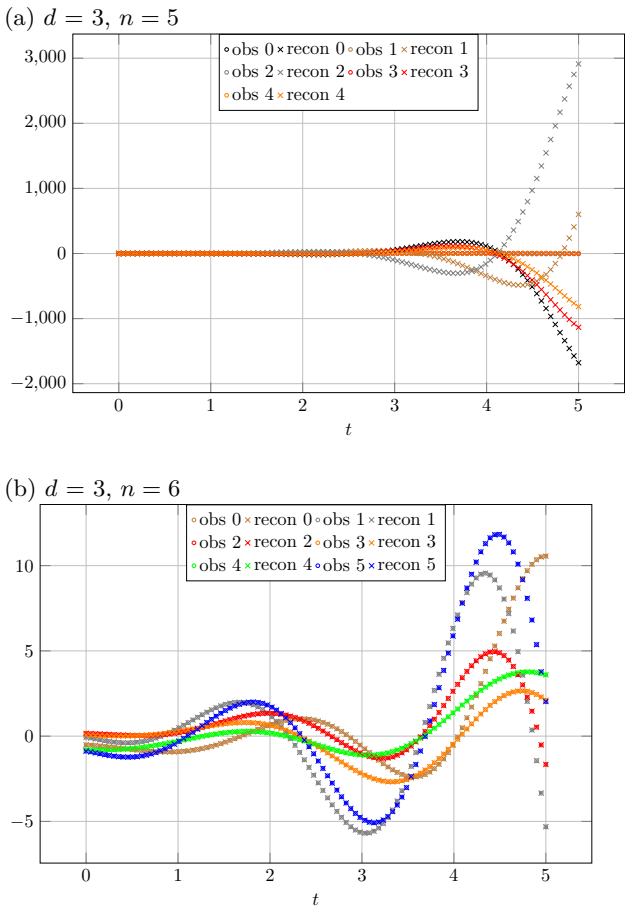

FIG. 7. Reconstructed and observed time-series data. A circle (∘) denotes an observed data point (*obs*), and a cross (×) denotes a reconstructed data point (*recon*). In the legend, the numbers following the type of data (i.e., *obs* or *recon*) denote the indices of the data. The parameters are as follows: (a) $d = 3$, $n = 5$, (b) $d = 3$, $n = 6$. For both panels, $\sigma = 10^{-4}$. The SVD rank r is set to the actual observation size n.

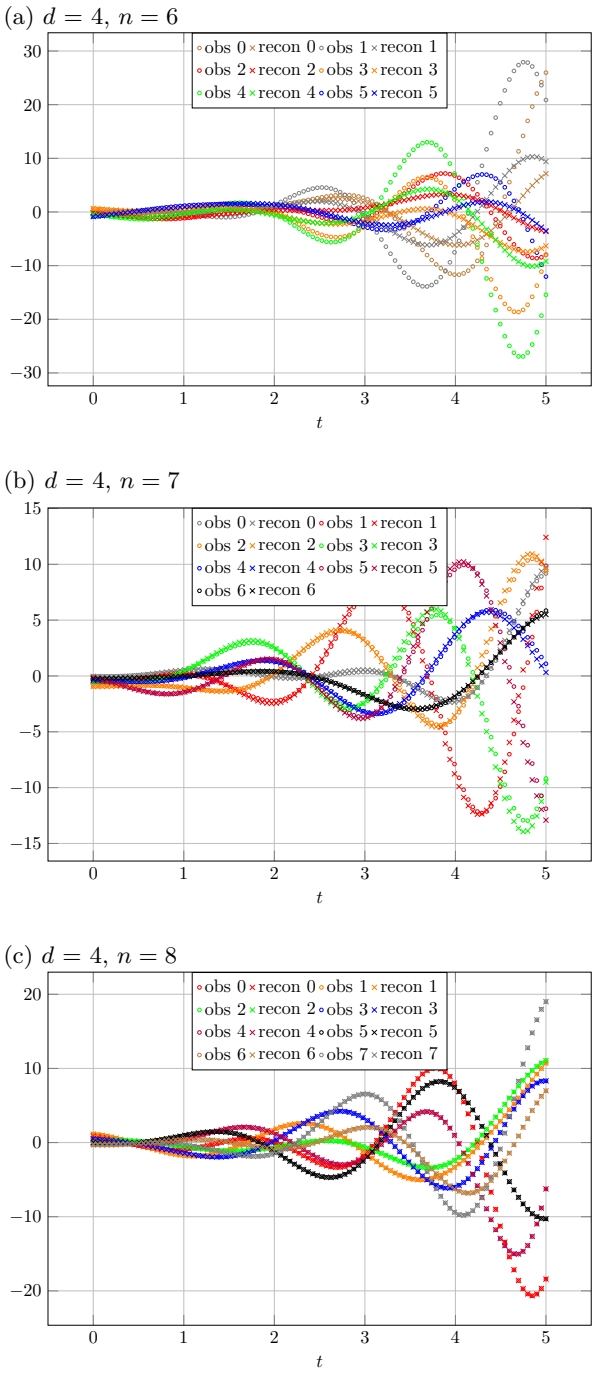

FIG. 8. Reconstructed and observed time-series data. A circle (∘) denotes an observed data point (*obs*), and a cross (×) denotes a reconstructed data point (*recon*). In the legend, the numbers following the type of data (i.e., *obs* or *recon*) denote the indices of the data. The parameters are as follows: (a) $d = 4$, $n = 6$, (b) $d = 4$, $n = 7$, (c) $d = 4$, $n = 8$. For all panels, $\sigma = 10^{-4}$. The SVD rank r is set to the actual observation size n.

Since the observation matrix R is randomly generated using Eq. (61), the reconstructed data sometimes suffers from huge reconstruction errors, resulting in the deviations of mean RMSEs from its trend in Fig. 6.

VIII. DISCUSSION

A fractional oscillator is an oscillator with power-law memory effects. The reconstruction of the frequencies in the previous section was consistent with the ground truth. The reconstruction error increased as a function of the noise standard deviation, whereas the dimension of the observation space did not significantly affect the error. Thus, we conclude that the fracDMD can reconstruct isolated fractional oscillators well.

We also performed numerical experiments for various values of α. The reconstruction error L has a minimum value at approximately $\alpha = \nu$ for the case without external forces with sufficiently weak noise. This implies that the proposed method is useful for estimating the order of the system equation. Noting that the case $\alpha = 1$ corresponds to DMD, we can also infer that our method achieved better reconstruction than DMD for fractional oscillators. Although this inference appears to be exact, the introduction of a higher-order scheme and fractional integral may lead to small discrepancies between the DMD and fracDMD results, even for $\alpha = 1$.

The cases with external forces need more careful consideration: the reconstructed frequencies and order may have a huge discrepancy from the ground truth, while the error is dependent on the time profile of the external force and the noise standard deviation σ. We have not conducted a comprehensive research on the effects of the external forces, however, it seems that the external forces only with low-frequency components lead to erroneous results. In the *impulse* case, the reconstruction error has its minima appropriately at the ground truth $\alpha = \nu$, meaning a successful estimation of the order α.

We performed numerical tests with partial observations using the fractional oscillators as well. We successfully reconstructed the partial observation data using the proposed method for sufficiently large observation dimensions. The minimum dimension of the observation needed to reconstruct the observed data approximately coincides with the intrinsic dimension of the system. This implies that the modeling of memory effects using the proposed method might be useful in various fields of science to extract the intrinsic dynamics from the observations. For example, in a macroscopic solid-state matter, we have around N_{A} (Avogadro's number) variables each of which describes the coordinate of an atom. However, the oscillation normal modes of the matter are usually highly degenerate and the dynamics of the oscillation is described by a small number of frequencies, enabling us to reconstruct the dynamics from a relatively small number ($\ll N_{\mathrm{A}}$) of observations.

Our method can be used to model unknown physical processes such as the mechanical motion of industrial machines and thermal systems. One possible manner of using this method is to estimate the memory effects of the system. If the reconstruction error by fracDMD has its minima at approximately $\alpha = 1$, it can be concluded that the system is memoryless; otherwise, the optimal value of α can be used to include the memory effects in the model.

Among the possible extensions, modifications to incorporate the nonlinearity to the fractional DMD scheme might be of utmost importance. Another way to include more complex situations is to use nonpower-law memory. In our current numerical setup, we used power-law memories that were mathematically shown to be equivalent to fractional-order equations of motion [14]. However, in principle, the memories in real data can decay according to any smooth function. Possible extensions are to use the Caputo-Fabrizio fractional derivative [30], or exponential-law memory [19]. We can use the Caputo-Fabrizio fractional derivative to circumvent the singularity around the origin of time difference, while the use of exponential-law memory might enable us to incorporate different types of physics.

We can further extend our method by using an arbitrary memory kernel function $g(t, t')$ in Eq. (15). Noting that the causal linear operator π has a one-to-one correspondence to the memory kernel such that $g(t, t') = g(t, t')\Theta(t - t')$, one can construct the corresponding matrix D_π using the method described in the main text. Although we can no longer enjoy the advantages of fractional derivatives (i.e., they form a one-parameter group, and easily estimated using the existing numerical libraries), this generalization might be quite useful in some cases where the power-law does not holds.

Within the scope of fractional derivative, another possible extension is to include multiple derivatives in the time-evolution model of $\boldsymbol{x}$, as follows:

$$\sum_{k=0}^{K} A_k \frac{\mathrm{d}^{\alpha+k}\boldsymbol{x}}{\mathrm{d}t^{\alpha+k}} = B\boldsymbol{u}(t), \tag{62}$$

where A_k is the coefficient matrix for $\alpha + k$-th order derivative, and B is the coefficient for the external force $\boldsymbol{u}(t)$. In this way, we might extend our method for the systems with fractional transfer functions. One famous example is the Warburg impedance, extensively studied in the context of lithium-ion batteries [31].

In the reconstruction considered herein, we assumed that we knew the actual size of the problem (i.e., system size). This assumption makes the problem easier to solve. However, in realistic situations, the actual dimensions of system equations are rarely known. In future work, we may optimize the rank of the system and coefficients.

IX. SUMMARY

We proposed a new framework DMDm (a DMD-based numerical tool) to analyze time-series data. The use of a more generic linear operator instead of a finite difference operator enabled us to consider the memory effects in time-evolution models. The memory effect is an extensively observed phenomenon in the real world, as observed in (among others) viscoelastic matter and fluid

dynamics. As an example of DMDm, we formulated fracDMD to indicate the use of a fractional-order derivative. The incorporation of a fractional-order derivative in DMDm is equivalent to assuming power-law memory effects. We successfully demonstrated that by using fracDMD, the frequencies of fractional oscillators can be reconstructed from noisy observations. The proposed method is expected to be useful for modeling unknown physical processes such as thermal and mechanical processes in modern industrial machines.

AUTHOR CONTRIBUTIONS

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

CONFLICT OF INTEREST

The authors declare that they have no conflicts of interest.

ACKNOWLEDGMENTS

We thank our teammates for their comments on the theoretical and practical aspects of the proposed method. Mr. Takashi Yokota and Mr. Masaru Nishimura of Tokyo Electron Ltd. provided valuable comments on the novelty of the proposed method. We also thank them for their aid in patent application filings of the proposed methods. Dr. Shota Yamada gave us insightful comments and examples of memory effects. Our survey of existing research was partly supported by the Access to Articles Program of the Tokyo Electron Group.

Appendix A: Improvements in Numerical Implementation

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

In appendix A 2, we present a scheme wherein the discretization error converges to zero with respect to second-order time intervals. A more accurate reconstruction was expected for the second-order scheme. In appendix A 3, we demonstrate the use of the fractional *integral* instead of the fractional derivative in fracDMD. In fractional calculus, it may be considered that the derivative and integral are treated in a unified manner; however, there are certain subtleties in the fractional derivative because the derivative of the order $\alpha > 1$ requires integer-order derivatives in the Caputo derivative. In certain limited cases, the use of fractional derivatives can be circumvented and hence, integer-order derivatives can be applied 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 $\Delta t > 0$.

1. First-order Numerical Scheme Revisited

We discretize the following equation within a *first-order* error in Δt, assuming $T = t_{m-1} - t_0$ is 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:

$$D_f^{(\alpha)}(t) = \frac{1}{\Gamma(\alpha)} \int_0^t \mathrm{d}\tau\, f(\tau)(t-\tau)^{\alpha-1}. \tag{A1}$$

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

$$\begin{aligned} &D_f^{(\alpha)}(t_{k+1}) \\ &= \frac{1}{\Gamma(\alpha)} \sum_{i=0}^{k} \int_{t_i}^{t_{i+1}} \mathrm{d}\tau\, (f(t_i) + \mathcal{O}(\Delta t))(t_{k+1} - \tau)^{\alpha-1} \\ &= \frac{1}{\Gamma(\alpha)\alpha} \sum_{i=0}^{k} f(t_i)((t_{k+1} - t_i)^{\alpha} - (t_{k+1} - t_{i+1})^{\alpha}) + \mathcal{O}(\Delta t), \end{aligned} \tag{A2}$$

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

$$\begin{aligned} D(t_{k+1}) = \frac{1}{\Gamma(\alpha+1)} \sum_{i=0}^{k} \left((t_{k+1} - t_i)^{\alpha} - (t_{k+1} - t_{i+1})^{\alpha}\right) f(t_i) \\ + \mathcal{O}(\Delta t). \end{aligned} \tag{A3}$$

In the matrix representation Eq. (43) in the main text, we require matrix $W^{(\alpha)} = \left[\boldsymbol{w}_0^{(\alpha)}, \boldsymbol{w}_1^{(\alpha)}, ..., \boldsymbol{w}_{m-1}^{(\alpha)}\right]$ to be

invertible. To achieve this, the derivative at $t = t_{k+1}$ can be stored in the k-th element of the resultant (row) vector $\vec{y} = \vec{x}W^{(\alpha)}$ for given time-series data $\vec{x}$ such that the diagonal elements of $W^{(\alpha)}$ become nonzero without violating the requirements of the first-order scheme. By adopting this convention, we can derive the expression for weights Eq. (44) in the main text.

2. Second-order Numerical Scheme

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

$$D_f^{(\nu)}(t) = \frac{1}{\Gamma(\nu)} \int_0^t \mathrm{d}\tau \, \frac{f(\tau)}{(t-\tau)^{1-\nu}}, \tag{A4}$$

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

$$\begin{aligned} D_f^{(\nu)}(t_{k+1}) &= D_f^{(\nu)}((k+1)\Delta t) \\ &= \frac{1}{\Gamma(\nu)} \int_0^{(k+1)\Delta t} \mathrm{d}\tau \, \frac{f(\tau)}{((k+1)\,\Delta t - \tau)^{1-\nu}}. \end{aligned} \tag{A5}$$

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

$$\begin{aligned} &\frac{1}{\Gamma(\nu)} \int_0^{(k+1)\Delta t} \mathrm{d}\tau \, \frac{f(\tau)}{((k+1)\,\Delta t - \tau)^{1-\nu}} \\ &= \frac{1}{\Gamma(\nu)} \sum_{i=0}^{k} \int_{i\Delta t}^{(i+1)\Delta t} \mathrm{d}\tau \, \frac{f(\tau)}{((k+1)\Delta t - \tau)^{1-\nu}} \\ &= \frac{1}{\Gamma(\nu)} \sum_{i=0}^{k} \int_0^1 \mathrm{d}s \, \Delta t \frac{f(i\Delta t + s\Delta t)}{((k+1)\Delta t - (i\Delta t + s\Delta t))^{1-\nu}} \\ &= \frac{1}{\Gamma(\nu)} \sum_{i=0}^{k} \int_0^1 \mathrm{d}s \, \Delta t \, ((k+1)\Delta t - (i+s)\Delta t)^{\nu-1} \, f(i\Delta t + s\Delta t). \end{aligned} \tag{A6}$$

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

$$f(i\Delta t + s\Delta t) = (1-s)f_i + sf_{i+1} + \mathcal{O}(\Delta t^2). \tag{A7}$$

Substituting Eq. (A7) into Eq. (A6), we obtain the following expression for $D_f^{(\nu)}(t_{k+1})$:

$$D_f^{(\nu)}(t_{k+1}) = \frac{1}{\Gamma(\nu)} \sum_{i=0}^{k} \int_0^1 \mathrm{d}s \, (\Delta t)^{\nu} ((k+1) - (i+s))^{\nu-1} [(1-s)f_i + sf_{i+1}] + \mathcal{O}(\Delta t^2). \tag{A8}$$

We introduce the *distance* between time points k, i as $C_{k,i} = k - i + 1$ and the normalized (inverse) distance $D_{k,i} = 1 - C_{k,i}^{-1} \in [0, 1)$. A straightforward calculation yields:

$$\begin{aligned} D_f^{(\nu)}(t_{k+1}) = \frac{1}{\Gamma(\nu+2)} \sum_{i=0}^{k} \Delta t^{\nu} C_{k,i}^{\nu+1} \left[f_i \left(D_{k,i}^{\nu+1} - D_{k,i} - \nu C_{k,i}^{-1} \right) + f_{i+1} \left(1 - D_{k,i}^{\nu} \left(1 + \nu C_{k,i}^{-1} \right) \right) \right] \\ + \mathcal{O}(\Delta t^2). \end{aligned} \tag{A9}$$

Subsequently, we attempt to express the summation in Eq. (A9) using the dot products of the 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 a weight vector.

$$\psi_i^{(k+1)} = \begin{cases} P_{k+1}^{(\nu)}(0) & (i=0) \\ P_{k+1}^{(\nu)}(i) + Q_{k+1}^{(\nu)}(i-1) & (1 \le i \le k) \\ Q_{k+1}^{(\nu)}(k) & (i=k+1), \end{cases} \tag{A10}$$

where,

$$\begin{aligned} P_{k+1}^{(\nu)}(i) &= \frac{1}{\Gamma(\nu+2)} \Delta t^{\nu} C_{k,i}^{\nu+1} \Big(D_{k,i}^{\nu+1} - D_{k,i} - \nu C_{k,i}^{-1} \Big), \\ Q_{k+1}^{(\nu)}(i) &= \frac{1}{\Gamma(\nu+2)} \Delta t^{\nu} C_{k,i}^{\nu+1} \Big[1 - D_{k,i}^{\nu} \Big(1 + \nu C_{k,i}^{-1} \Big) \Big]. \end{aligned} \tag{A11}$$

The value of the fractional derivative $D_f^{(\nu)}$ at time $t = t_{k+1}$ was estimated using the following expression:

$$D_f^{(\nu)}(t_{k+1}) = \big[f_0, f_1, ..., f_{k+1} \big] \boldsymbol{\psi}^{(k+1)} + \mathcal{O}(\Delta t^2). \tag{A12}$$

Note that Eq. (A12) is the second-order scheme for Eq. (A4). To implement the second-order numerical scheme for fracDMD, the weight vector $\boldsymbol{w}_k^{(\nu)}$ in Eq. (44) can be simply replaced by the following expression:

$$(\boldsymbol{w}_k^{(\nu)})_i = \begin{cases} \psi_i^{(k)} & (i \le k) \\ 0 & (i > k). \end{cases} \tag{A13}$$

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

3. Use of the Fractional Integral

In the main text, we derived fracDMD using the fractional differential equation Eq. (48), wherein 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:

$$\frac{\mathrm{d}^{\alpha} \boldsymbol{x}}{\mathrm{d}t^{\alpha}} = A\boldsymbol{x} + B\boldsymbol{u}(t). \tag{A14}$$

However, a naïve discretization of the original form Eq. (A14) leads to subtlety in the numerical treatments at approximately $t = 0$, and the solution of the time-evolution equation Eq. (48) 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 an inconsistency at $t = 0$. This is the primary reason for the assumption that $\pi \in \hom(\mathbf{C}^{\mathcal{T}}, \mathbf{C}^{\mathcal{S}})$ with $\mathcal{S} = (0, T] \subset [0, T] = \mathcal{T}$ for linear operator π in the main text.

Alternatively, to circumvent this difficulty, we can perform a fractional derivative of order $-\alpha$ on both sides of equation Eq. (A14) to *cancel* the derivative on the left-hand side and obtain an alternative form for $\alpha < 0$ as follows:

$$\boldsymbol{x} \stackrel{?}{=} \frac{\mathrm{d}^{-\alpha}}{\mathrm{d}t^{-\alpha}} (A\boldsymbol{x} + B\boldsymbol{u}(t)). \tag{A15}$$

We can demonstrate that this transformation is accurate for $\alpha < 0$ [23]. However, for an arbitrary integrable function $\boldsymbol{x} = \boldsymbol{x}(t)$ and $\alpha > 0$, the transformation from the original form Eq. (A14) to the alternative form Eq. (A15) is not possible because, in general, the function $\boldsymbol{x}$ has nonzero integer-order derivatives at $t = 0$. However, it can be shown that the difference between the original function $\boldsymbol{x}$ and the retrieved function $\tilde{\boldsymbol{x}} = \frac{\mathrm{d}^{-\alpha}}{\mathrm{d}t^{-\alpha}} \frac{\mathrm{d}^{\alpha} \boldsymbol{x}}{\mathrm{d}t^{\alpha}}$ can be expressed as a polynomial of t, as (see Lemma 2.22 in [23]),

$$\frac{\mathrm{d}^{-\alpha}}{\mathrm{d}t^{-\alpha}} \frac{\mathrm{d}^{\alpha} \boldsymbol{x}}{\mathrm{d}t^{\alpha}} = \boldsymbol{x}(t) - \sum_{\ell=0}^{n_{\alpha}^{+}-1} \frac{\boldsymbol{x}^{(\ell)}(0)}{\ell!} t^{\ell}. \tag{A16}$$

Because $\alpha = 0$ is a trivial case, mode decomposition can be performed by discretizing the alternative form Eq. (A15) for $\alpha < 1$. The resultant equation is as follows:

$$\boldsymbol{x} = \frac{\mathrm{d}^{-\alpha}}{\mathrm{d}t^{-\alpha}} (A\boldsymbol{x} + B\boldsymbol{u}(t)) + \boldsymbol{x}(0). \tag{A17}$$

Even for $\alpha > 1$, Eq. (A17) is an approximation of the exact fractional integral equation Eq. (A16). An alternative version of fracDMD uses Eq. (A17) and the same procedure described in the main text.

Appendix B: Numerical Test

In Section VIII, we used the approximated alternative form of fracDMD shown in the previous section (i.e., the fractional integral equation Eq. (A17) and second-order discretization Eq. (A13)).

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

[2] 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**, 391 (2014).

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

[4] P. J. Schmid, Dynamic mode decomposition and its variants, Annual Review of Fluid Mechanics **54**, 225 (2022).

[5] E. Rodrigues, B. Zadrozny, C. Watson, and D. Gold, Decadal forecasts with resdmd: a residual dmd neural

network, arXiv preprint arXiv:2106.11111 (2021).
[6] 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**, 20210199 (2022).
[7] 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**, 112001 (2019).
[8] 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**, 1 (2021).
[9] I. Mezic, Analysis of fluid flows via spectral properties of the koopman operator, Annual Review of Fluid Mechanics **45**, 357 (2013).
[10] I. Mezić, Spectral properties of dynamical systems, model reduction and decompositions, Nonlinear Dynamics **41**, 309 (2005).
[11] C. W. Rowley, I. Mezić, S. Bagheri, P. Schlatter, and D. S. Henningson, Spectral analysis of nonlinear flows, Journal of fluid mechanics **641**, 115 (2009).
[12] J. L. Proctor, S. L. Brunton, and J. N. Kutz, Dynamic mode decomposition with control, SIAM Journal on Applied Dynamical Systems **15**, 142 (2016).
[13] T. Askham and J. N. Kutz, Variable projection methods for an optimized dynamic mode decomposition, SIAM Journal on Applied Dynamical Systems **17**, 380 (2018).
[14] A. Svenkeson, B. Glaz, S. Stanton, and B. J. West, Spectral decomposition of nonlinear systems with memory, Phys. Rev. E **93**, 022211 (2016).
[15] S. Burov and E. Barkai, Fractional langevin equation: Overdamped, underdamped, and critical behaviors, Physical Review E **78**, 031112 (2008).
[16] F. Al-Bender, V. Lampaert, and J. Swevers, The generalized maxwell-slip model: a novel model for friction simulation and compensation, IEEE Transactions on automatic control **50**, 1883 (2005).
[17] M. Rakotondrabe, Bouc–wen modeling and inverse multiplicative structure to compensate hysteresis nonlinearity in piezoelectric actuators, IEEE Transactions on Automation Science and Engineering **8**, 428 (2011).
[18] M. Rakotondrabe, C. Clevy, and P. Lutz, Complete open loop control of hysteretic, creeped, and oscillating piezoelectric cantilevers, IEEE Transactions on Automation Science and Engineering **7**, 440 (2010).
[19] D. Panja, Anomalous polymer dynamics is non-markovian: memory effects and the generalized langevin equation formulation, Journal of Statistical Mechanics: Theory and Experiment **2010**, P06011 (2010).
[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**, 1446 (1995).
[21] H. Lei, N. A. Baker, and X. Li, Data-driven parameterization of the generalized langevin equation, Proceedings of the National Academy of Sciences **113**, 14183 (2016).
[22] V. E. Tarasov, General fractional dynamics, Mathematics **9**, 1464 (2021).
[23] A. A. Kilbas, H. M. Srivastava, and J. J. Trujillo, *Theory and applications of fractional differential equations*, Vol. 204 (elsevier, 2006).
[24] A. A. K. Samko, Stefan G. and O. I. Marichev, *Fractional integrals and derivatives: theory and applications* (Gordon and Breach Science Publishers, 1993).
[25] 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.
[26] I. Podlubny, Fractional-order systems and pi/sup /spl lambda//d/sup /spl mu//-controllers, IEEE Transactions on Automatic Control **44**, 208 (1999).
[27] H. J. Haubold, A. M. Mathai, and R. K. Saxena, Mittag-leffler functions and their applications, Journal of applied mathematics **2011** (2011).
[28] 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.
[29] A. A. Stanislavsky, Fractional oscillator, Phys. Rev. E **70**, 051103 (2004).
[30] M. Caputo and M. Fabrizio, A new definition of fractional derivative without singular kernel. prog fract differ appl 1 (2): 73–85 (2015).
[31] A. Baba and S. Adachi, Simultaneous state of charge and parameter estimation of lithium-ion battery using log-normalized unscented kalman filter, in *2015 American Control Conference (ACC)* (2015) pp. 311–316.