PLOS ONE

RESEARCH ARTICLE

Extended-Kalman-filter-based dynamic mode decomposition for simultaneous system identification and denoising

Taku Nonomura[1,2]*, **Hisaichi Shibata**[3], **Ryoji Takaki**[3]

1 Department of Aerospace Engineering, Graduate School of Engineering, Tohoku University, Sendai, Miyagi, Japan, **2** Presto, JST, Sendai, Miyagi, Japan, **3** Institute of Space and Astronautical Science, Japan Aerospace Exploration Agency, Sagamihara, Kanagawa, Japan

* nonomura@aero.mech.tohoku.ac.jp

Citation: Nonomura T, Shibata H, Takaki R (2019) Extended-Kalman-filter-based dynamic mode decomposition for simultaneous system identification and denoising. PLoS ONE 14(2): e0209836. https://doi.org/10.1371/journal.pone.0209836

Editor: Baogui Xin, Shandong University of Science and Technology, CHINA

Received: May 8, 2018

Accepted: December 12, 2018

Published: February 21, 2019

Data Availability Statement: All relevant data are within the paper and its Supporting Information files.

Funding: This work was supported by PREST, JST (JPMJPR1678). URL: https://www.jst.go.jp/kisoken/presto/en/index.html. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Competing interests: The authors have declared that no competing interests exist.

Abstract

A new dynamic mode decomposition (DMD) method is introduced for simultaneous system identification and denoising in conjunction with the adoption of an extended Kalman filter algorithm. The present paper explains the extended-Kalman-filter-based DMD (EKFDMD) algorithm which is an online algorithm for dataset for a small number of degree of freedom (DoF). It also illustrates that EKFDMD requires significant numerical resources for many-degree-of-freedom (many-DoF) problems and that the combination with truncated proper orthogonal decomposition (trPOD) helps us to apply the EKFDMD algorithm to many-DoF problems, though it prevents the algorithm from being fully online. The numerical experiments of a noisy dataset with a small number of DoFs illustrate that EKFDMD can estimate eigenvalues better than or as well as the existing algorithms, whereas EKFDMD can also denoise the original dataset online. In particular, EKFDMD performs better than existing algorithms for the case in which system noise is present. The EKFDMD with trPOD, which unfortunately is not fully online, can be successfully applied to many-DoF problems, including a fluid-problem example, and the results reveal the superior performance of system identification and denoising.

1 Introduction

Recently, modal decomposition [1] for fluid dynamics has attracted attention from the viewpoints of data reduction, data analysis, and reduced-order modeling of complex dataset. This is one method for data-driven science in fluid dynamics. The most conventional method of modal decomposition is a proper orthogonal decomposition (POD), [2, 3] which is also called principal component analysis (PCA) and Karhunen-Loéve expansion. The standard POD can be computed by singular value decomposition (SVD), and this fact explains that the obtained modes are orthogonal with respect to each other. Proper orthogonal decomposition modes can be computed by snapshots of fluid data and can be used for both numerical and experimental approaches. Based on POD modes, a reduced-order model can be constructed with the

Galerkin projection method for instance, although only a numerical approach can be used for reduced-order modeling in this way.

Another conventional method is global linear stability analysis (GLSA), [4–6] which shows that the eigenmodes of the system of linearized governing equations (i.e., the Navier-Stokes equations for most of the fluid problems) around the steady state of nonlinear dynamics. Here, GLSA shows the most unstable eigenmodes and judges whether the steady-state solution is stable. The modes obtained by GLSA are a solution of the original linearized equations, although this method always requires numerically complex approaches and cannot be applied to experimental data. Unlike POD modes, the modes obtained by GLSA are not orthogonal unless otherwise the system is written with an Hermite operator.

In recent decades, a new method, dynamic mode decomposition (DMD), [7] has been proposed and developed as a data-driven science method and has been applied to numerous fluid problems. [8–11] Here, DMD has characteristics of both POD and GLSA, whereas DMD can be computed only by a time-series of snapshots of numerical and experimental data. This method processes snapshots of sequential unsteady nonlinear flow fields and yields eigenvalues and corresponding eigenmodes for the case in which the dataset is assumed to be explained by a linear system $\boldsymbol{x}_{k+1} = A\boldsymbol{x}_k$, where x_k is the kth snapshot of sequential data and A is a system matrix. These dynamic modes are generally nonorthogonal, and each mode possesses a single-frequency response with amplification or damping as a natural characteristic of a linear system expression, which leads to a more intrinsic understanding of the role of each mode. Thus far, there are several methods by which to compute the dynamic modes: standard DMD [7], exact DMD, noise-cancelling DMD (ncDMD), [12] forward-backward DMD, (fbDMD), [12] total least-squares DMD (tlsDMD), [12, 13] online DMD, [14] and Kalman-filter-based DMD (KFDMD), [15] where ncDMD, fbDMD, tlsDMD and KFDMD focus on the noisy dataset. The standard DMD and the exact DMD adopt SVD and a Moore-Penrose pseudo-inverse matrix for low-rank approximation of the matrix A, respectively. This implies that these algorithms compute dynamic modes as a kind of least-squares problem. A robust method for a noisy dataset, tlsDMD, adopts a truncated POD for pair data and successfully increases the accuracy of obtained dynamic modes. A recent KFDMD is written in the form of system identification using the Kalman filter algorithm [16] and can be optimized based on the prior knowledge of the noise superimposed on the data. This is different from the usage of the Kalman filter in Reference [17, 18] in which the Kalman filter is used for data reconstruction and prediction.

However, the application of DMD to noisy data and the denoising process are still limited. For example, tlsDMD has been developed for accurately estimating the dynamic modes and corresponding eigenvalues, but a method by which to reconstruct the data has rarely been shown except for the data reconstruction using first snapshot, [19] which is conventionally adopted. If we adopt the conventional simple estimation of initial amplitudes to reconstruct the data, then the data is greatly affected by the noise on the initial data, as expected. One of a few advanced data reconstruction methods is use of Kalman filter for linear system that is corresponding to the Koopman operator after the linear system is estimated. [17, 18]

Optimized DMD (optDMD), [20] and the combination of tlsDMD [12, 21] and the sparsity-promoting DMD (spDMD) [22] could be used for the denoising of noisy data. Here, optDMD gives us the dynamic modes and eigenvalues and corresponding initial values that best fit the noisy time series data under the assumption of no system noise. On the other hand, spDMD [22] selects finite-number modes for the reconstruction of flow fields considering the L_0 or L_1 norm of regularization terms, as is often used in sparse modeling and compressed sensing. These methods are very useful for reconstructing flow fields, but the reconstructed data are governed by the initial value of the strength of each mode and possibly cannot handle

the change in phase of dynamic modes in long-time data due to the system noise including modeling error, nonlinear processes, or unexpected events in the experiments. Furthermore, the optDMD requires fitting of all of the data, and the combination procedure of tlsDMD and spDMD requires two-step computation. At present, an online method for simultaneous system identification and denoising using the DMD framework has not yet been proposed.

In the present paper, a new method for simultaneous system identification and denoising using the DMD framework is proposed using the extended Kalman filter. In addition to the system identification of the previously proposed KFDMD, [15] the observed data are simultaneously filtered online for dataset with a small number of degree of freedom (DoF). The present paper first explains the algorithm of the proposed extended-Kalman-filter-based DMD (EKFDMD). The drawback of the computational costs of EKFDMD is addressed, and combination with a truncated POD (trPOD) is proposed for reduction of the computational cost, though it prevents the algorithm from being fully online. Finally, the proposed method is applied to various problems and its performance is illustrated.

2 Previous methods compared in the present study

2.1 Problem settings

Here, the previous algorithms compared in the present study are briefly explained. For the extension in the next subsection, the linear system model is assumed for the time series dataset as follows:

$$\boldsymbol{x}_{k+1} = A\boldsymbol{x}_k + \boldsymbol{v}_k, \tag{1}$$

$$\boldsymbol{y}_k = \boldsymbol{x}_k + \boldsymbol{w}_k. \tag{2}$$

Here, A, $\boldsymbol{x}$, $\boldsymbol{y}$, $\boldsymbol{v}$, $\boldsymbol{w}$ and a subscript k are the system matrix, the state variable vector, the observation vector, the system noise, the observation noise, and the time step respectively. Here, the dimension of the state and observed variables is set to be n. Moreover, $\boldsymbol{x}_k$ is assumed to be the true value. Usually, we can only access $\boldsymbol{y}$ in the present paper, though $\boldsymbol{x}$ has been used as the observation vector in the previous DMD studies. Therefore, the reader should take care when considering the notation used herein. First, three methods, DMD, tlsDMD, and KFDMD are briefly explained in Subsections 2.2, 2.3, and 2.4, respectively, and a conventional data reconstruction method for these algorithms is introduced in Subsection 2.5. Finally, optDMD, which is a state-of-art offline algorithm for both estimating the dynamic modes and reconstructing data, is explained in Subsection 2.6.

2.2 DMD

The m-sample observation data matrix including observation noise is defined as follows:

$$Y_{1:m} = (\boldsymbol{y}_m, \boldsymbol{y}_{m-1}, \cdots, \boldsymbol{y}_2, \boldsymbol{y}_1), \tag{3}$$

whereas $\boldsymbol{y}_k = \boldsymbol{x}_k$ if the observation noise is absent. The original DMD is performed with SVD for $Y_{1:m-1}$ as follows:

$$Y_{1:m-1} = U_{1:m-1}\Sigma_{1:m-1}V^{\mathrm{T}}_{1:m-1}. \tag{4}$$

Here, U, Σ, and V are a left singular matrix, a diagonal matrix with singular values, and a right singular matrix, respectively. As described in the original DMD paper, a truncated POD (SVD) is used to filter the noise. Therefore, the rank r approximation of the observation data

matrix is obtained as follows:

$$Y_{1:m-1} \sim \tilde{U}_{1:m-1}\tilde{\Sigma}_{1:m-1}\tilde{V}^{\mathrm{T}}_{1:m-1}. \tag{5}$$

In this case, the projected $r \times r$ matrix $\tilde{A}$ of the matrix A onto the low-dimensional space can be obtained as follows:

$$\tilde{A} = \tilde{U}^{\mathrm{T}}_{1:m-1} Y_{2:m} \tilde{V}_{1:m-1}\tilde{\Sigma}^{-1}_{1:m-1}. \tag{6}$$

Then, the eigendecomposition is carried out:

$$\tilde{A}\, W_{\mathrm{DMD}} = W_{\mathrm{DMD}}\Lambda_{\mathrm{DMD}}. \tag{7}$$

Here, W_{DMD} are the eigenvectors, and Λ_{DMD} is the diagonal matrix with the eigenvalues. Using W_{DMD}, the dynamic mode matrix in the original space is recovered:

$$\Phi = \tilde{U}_{1:m-1} W_{\mathrm{DMD}}. \tag{8}$$

Here, Φ contains the dynamic mode vectors as follows:

$$\Phi = [\phi_1 \quad \phi_2 \quad \dots \quad \phi_r]. \tag{9}$$

2.3 tlsDMD

For total least-squares DMD, the pair snapshot is considered. In this case, trPOD data or raw data can be used. [12, 13] In the present study, raw data are directly used as in the original code. [23] The procedure for the time series data are as follows. First, define a pair data matrix:

$$Z = \begin{bmatrix} Y_{1:m-1} \\ Y_{2:m} \end{bmatrix} \tag{10}$$

and POD is applied to the pair data matrix above:

$$Z = \begin{bmatrix} U_{1:m-1} \\ U_{2:m} \end{bmatrix} \Sigma_Z V_Z^{\mathrm{T}}. \tag{11}$$

Then, we obtain an r-rank truncated pair POD, as follows:

$$\hat{Z} \sim \begin{bmatrix} \hat{U}_{1:m-1} \\ \hat{U}_{2:m} \end{bmatrix} \hat{\Sigma}_Z \hat{V}_Z^{\mathrm{T}}. \tag{12}$$

Here, we obtain a snapshot pair of POD projections $\hat{Y}_{1:m-1}$ and $\hat{Y}_{2:m}$ of $Y_{1:m-1}$ and $Y_{2:m}$ as follows:

$$\hat{Y}_{1:m-1} = Y_{1:m-1}\hat{V}_Z \tag{13}$$

$$\hat{Y}_{2:m} = Y_{2:m}\hat{V}_Z. \tag{14}$$

Using these matrices, $\tilde{A}$ is computed by SVD of $\hat{Y}_{1:m-1}$:

$$\hat{Y}_{1:m-1} = U_{\hat{Y}_{1:m-1}} \Sigma_{\hat{Y}_{1:m-1}} V_{\hat{Y}_{1:m-1}}, \tag{15}$$

$$\tilde{A} = U^{\mathrm{T}}_{\hat{Y}_{1:m-1}} \hat{Y}_{2:m} V_{\hat{Y}_{1:m-1}} \Sigma^{-1}_{\hat{Y}_{1:m-1}}. \tag{16}$$

The dynamic mode and eigenvalue estimations are estimated exactly in the same way as DMD in Eqs 7 to 9.

2.4 KFDMD

The components of the matrix A are considered to be state variables of the Kalman filter. The state variable vector $\boldsymbol{\theta}^{\mathrm{KF}}$ are written as follows:

$$\boldsymbol{\theta}^{\mathrm{KF}} = \mathrm{vec}(A^{\mathrm{T}}). \tag{17}$$

Using the state variable vector described above, the system and observation equations can be written as follows:

$$\boldsymbol{\theta}^{\mathrm{KF}}_{k+1} = \boldsymbol{\theta}^{\mathrm{KF}}_{k} + \boldsymbol{v}_k, \tag{18}$$

$$\boldsymbol{y}_k = H^{\mathrm{KF}}_k \boldsymbol{\theta}^{\mathrm{KF}}_k + \boldsymbol{w}_k, \tag{19}$$

where H^{KF}_k is the following observation matrix defined as follows:

$$H^{\mathrm{KF}}_k = \overbrace{\begin{bmatrix} \boldsymbol{y}^{\mathrm{T}}_{k-1} & \mathbf{0} & \cdots & \cdots & \mathbf{0} \\ \mathbf{0} & \boldsymbol{y}^{\mathrm{T}}_{k-1} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \ddots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \cdots & \mathbf{0} & \boldsymbol{y}^{\mathrm{T}}_{k-1} & \mathbf{0} \\ \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \boldsymbol{y}^{\mathrm{T}}_{k-1} \end{bmatrix}}^{n^2 \text{ dimensions}} \Bigg\} n \text{ dimensions} \tag{20}$$

Note that we have the following relationship:

$$H^{\mathrm{KF}}_k \boldsymbol{\theta}^{\mathrm{KF}}_k = A_k \boldsymbol{y}_{k-1}, \tag{21}$$

where A_k represents the estimation of A in the kth time step. Here, $\boldsymbol{v}_k$ and $\boldsymbol{w}_k$ are system and observation noises, respectively. Using the state equation given above, the linear Kalman filter is constructed with the fast algorithm shown in Reference. [15] After obtaining the matrix A, the dynamic mode and corresponding eigenvalues are obtained through the eigendecomposition of the matrix A.

2.5 Data reconstruction using DMD, tlsDMD, and KFDMD

The DMD, tlsDMD, and KFDMD methods only estimate the matrix A and do not estimate the reconstructed time series data using dynamic modes. In a conventional method [19] of reconstruction, we assume that the data can be reconstructed as follows:

$$X_{\mathrm{reconst}} = \Phi B_0 V_{\mathrm{and}}, \tag{22}$$

Here, X_{reconst} is the reconstructed data matrix, B_0 is a diagonal matrix of the initial amplitudes

b_i of dynamic modes Φ_i, where

$$B_0 = \begin{bmatrix} b_1 & 0 & 0 & 0 \\ 0 & b_2 & 0 & 0 \\ 0 & 0 & \ddots & \vdots \\ 0 & 0 & \cdots & b_r \end{bmatrix}, \tag{23}$$

and V_{and} is a Vandermonde matrix representing the temporal behaviors of dynamic modes while assuming the system noise to be absent:

$$V_{\text{and}} = \begin{bmatrix} 1 & \lambda_1 & \lambda_1^2 & \cdots & \lambda_1^m \\ 1 & \lambda_2 & \lambda_2^2 & \cdots & \lambda_2^m \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ 1 & \lambda_r & \lambda_r^2 & \cdots & \lambda_r^m \end{bmatrix}. \tag{24}$$

The initial value vector $\boldsymbol{b}_0$, which is defined as

$$\boldsymbol{b}_0 = [b_1 \quad b_2 \quad \dots \quad b_r]^{\mathrm{T}}, \tag{25}$$

can be obtained using the pseudoinverse of Φ, as follows:

$$\boldsymbol{b}_0 = \Phi^{+}\boldsymbol{y}_0, \tag{26}$$

where the plus symbol superscript denotes the Moore-Penrose pseudoinverse matrix. As discussed later herein, $\boldsymbol{y}_0$ includes the observation noise superimposed on the initial snapshot, and this reconstruction does not work well due to this noise, even if the eigenvalues are well estimated.

2.6 optDMD

In the optimized DMD, [20] the following problem is solved:

$$[\Phi \quad B_0 \quad \Lambda] = \underset{\Phi,B,\Lambda}{\operatorname{argmin}} \parallel Y - X_{\text{reconst}} \parallel_F^2 \tag{27}$$

$$= \underset{\Phi,B,\Lambda}{\operatorname{argmin}} \parallel Y - \Phi B_0 V_{\text{and}} \parallel_F^2 . \tag{28}$$

Although there are several ways to solve this nonlinear problem above, the variable projection method is adopted in the present study. In this case, the best-fit reconstructed data matrix is obtained under the assumption that system noise is absent. In the case of spDMD, Φ and Λ are fixed using another DMD method, and optimum sparse $\boldsymbol{b}_0$ is solved while adding the L_1 or L_0 regularization term of $\boldsymbol{b}_0$. The original code [24] is employed in the present study.

3 Extended Kalman filter DMD

3.1 Algorithm

As introduced in the section above, we consider the system expressed by Eqs 1 and 2. For simplicity, we introduce Einstein summation convention for Eqs 1 and 2, as follows:

$$x_{i,k+1} = a_{ij}x_{j,k} + v_{i,k}, \tag{29}$$

$$y_{i,k} = x_{i,k} + w_{i,k}, \tag{30}$$

where $A = (a_{ij})$, $\boldsymbol{x} = (x_i)$, $\boldsymbol{y} = (y_i)$, $\boldsymbol{v} = (v_i)$, and $\boldsymbol{w} = (w_i)$.

Then, the Kalman filter algorithm is considered. In this problem, we would like to simultaneously conduct the online system identification and denoising of the observed variable when a number of DoF is small. Therefore, the observed variables and elements of the matrix A are chosen as state variables of the considered system. The state variable vector $\boldsymbol{\theta}$ is defined as follows:

$$\boldsymbol{\theta}_k = \begin{pmatrix} \boldsymbol{x}_k \\ \mathrm{vec}(A^{\mathrm{T}}) \end{pmatrix} = \left.\begin{pmatrix} x_{1,k} \\ x_{2,k} \\ \cdots \\ x_{n,k} \\ a_{11} \\ a_{12} \\ \vdots \\ a_{1n} \\ a_{21} \\ a_{22} \\ \vdots \\ a_{nn} \end{pmatrix}\right\} n + n^2 \text{ dimensions.} \tag{31}$$

Using these state variables, the system transient can be written as follows:

$$\boldsymbol{\theta}_{k+1} = \begin{pmatrix} \boldsymbol{x}_{k+1} \\ \mathrm{vec}(A^{\mathrm{T}}) \end{pmatrix} = \boldsymbol{f}(\boldsymbol{\theta}_k) = \boldsymbol{f}(\boldsymbol{x}_k, A) + \boldsymbol{v}_k, \tag{32}$$

$$\boldsymbol{y}_k = H\boldsymbol{\theta}_k + \boldsymbol{w}_k, \tag{33}$$

where the $\boldsymbol{v}_k$ and $\boldsymbol{w}_k$ are the system and observation noise, respectively, and the nonlinear

function $\boldsymbol{f}$ and the observation matrix are expressed as follows:

$$\boldsymbol{f} = \begin{pmatrix} A\boldsymbol{x}_k \\ \mathrm{vec}(A^{\mathrm{T}}) \end{pmatrix} = \begin{pmatrix} a_{1j}x_{j,k} \\ a_{2j}x_{j,k} \\ \cdots \\ a_{nj}x_{j,k} \\ a_{11} \\ a_{12} \\ \vdots \\ a_{1n} \\ a_{21} \\ a_{22} \\ \vdots \\ a_{nn} \end{pmatrix}, \quad H = \overbrace{(I \quad \mathbf{0})}^{n+n^2 \text{ dimensions}}\}. \; n \text{ dimensions} \tag{34}$$

The upper half of the system is written as the multiplication of state variables x_j and a_{ij}, and, as such, the system is considered to be nonlinear. The lower half of the system corresponds to the constant or slowly varying system coefficients to be identified and does not change explicitly. For the construction of the extended Kalman filter, the linearization is required. The Jacobian matrix F of a nonlinear function $\boldsymbol{f}$ of the state variables $\boldsymbol{\theta}$ is calculated as follows:

$$F_k = \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{\theta}_k} = \begin{pmatrix} \frac{\partial A\boldsymbol{x}_k}{\partial \boldsymbol{x}_k} & \frac{\partial A\boldsymbol{x}_k}{\partial \mathrm{vec}(A^{\mathrm{T}})} \\ \frac{\partial \mathrm{vec}(A^{\mathrm{T}})}{\partial \boldsymbol{x}_k} & \frac{\partial \mathrm{vec}(A^{\mathrm{T}})}{\partial \mathrm{vec}(A^{\mathrm{T}})} \end{pmatrix} = \overbrace{\begin{pmatrix} \boldsymbol{x}_k^{\mathrm{T}} & & \mathbf{0} \\ & \boldsymbol{x}_k^{\mathrm{T}} & \\ A & & \ddots \\ & & \\ \mathbf{0} & & \boldsymbol{x}_k^{\mathrm{T}} \\ \mathbf{0} & \boldsymbol{I} & \end{pmatrix}}^{n+n^2 \text{ dimensions}}\Bigg\}\, n+n^2 \text{ dimensions} \tag{35}$$

Using matrices F_k and H, the extended Kalman filter can be constructed for the nonlinear system. Note that F_k is a time-varying matrix.

Following the theory of a Kalman filter, a priori prediction of a state variable vector $\boldsymbol{\theta}_k$ and a covariance matrix $P_{k|k-1}$ can be achieved using the state variable vector $\boldsymbol{\theta}_k$ and covariance matrix $P_{k-1|k-1}$ from the previous time step,

$$\boldsymbol{\theta}_{k|k-1} = \boldsymbol{f}(\boldsymbol{\theta}_{k-1|k-1}), \tag{36}$$

$$P_{k|k-1} = F_k P_{k-1|k-1} F_k^{\mathrm{T}} + Q_k, \tag{37}$$

where the system matrix F_k is expressed by Eq 35, and Q is a covariance matrix of the system noise.

When a new observation is available, the state variables and the covariance matrix are updated using the Kalman gain, which is computed as

$$K_k = P_{k|k-1}H^{\mathrm{T}}S_k^{-1}, \tag{38}$$

where S_k is a noise covariance matrix and is expressed as follows:

$$S_k = R_k + HP_{k|k-1}H^{\mathrm{T}}. \tag{39}$$

Here, R_k is a covariance matrix of observation noise $\boldsymbol{w}_k$.

A modification vector for state variables $\boldsymbol{\theta}$ is computed as follows:

$$\delta\boldsymbol{\theta}_{k|k} = K_k(\boldsymbol{y}_k - H\boldsymbol{\theta}_{k|k-1}) \tag{40}$$

$$(= K_k(\boldsymbol{y}_k - A_{k|k-1}\boldsymbol{x}_{k-1|k-1})). \tag{41}$$

Finally, the state variable vector and the covariance matrix after the observation are updated as follows:

$$\boldsymbol{\theta}_{k|k} = \boldsymbol{\theta}_{k|k-1} + \delta\boldsymbol{\theta}_{k|k}, \tag{42}$$

$$P_{k|k} = (I - K_kH)P_{k|k-1}. \tag{43}$$

This extended Kalman filter requires the multiplication of the large matrix of dimension of $(n^2 + n) \times (n^2 + n)$, as discussed in Section 5. This is a clear drawback of this formulation for many-degree-of-freedom (many-DoF) problems, and using this algorithm together with trPOD is recommended, as explained in Section 3.2. This drawback of EKFDMD is the same as that of KFDMD designed for only the system identification, though the drawback of KFDMD is somehow relaxed owing to the fast algorithm proposed in the previous study, [15] in which the large matrix is assumed to be decomposed into several identical block matrices. Although we attempt to use a concept similar to the previous KFDMD, [15] we could not find a similar method for EKFDMD in the present state. Therefore, the computational cost for EKFDMD is severer than that for KFDMD designed for only system identification, and the use of the present algorithm together with trPOD is strongly recommended for many-DoF problems.

It should be noted that, in the early implementation of EKFDMD, we employed the several initial time steps for only the estimation of A without filtering of $\boldsymbol{x}$, but they are found to just degrade the results. In the present implementation, the simultaneous estimation is impulsively started from the first step.

3.2 Combination with a truncated POD

As discussed in the previous section, the computational cost of the present algorithm is high, and, therefore, a truncated POD (truncated SVD) should be used for the reduction in the number of DoFs of the dataset of the observed variables. Similar to a previous study on KFDMD for only system identification, the obtained data are processed as follows:

1. the batch POD is applied,
2. a proposed Kalman filter is then applied to the amplitude of each POD mode, and
3. the mode shape of a fluid system is finally recovered by multiplying the spatial POD modes.

As the first step (step 1), POD is applied to an observed data matrix and an observed data matrix is expressed in SVD form as follows:

$$Y_{1:m} = U_{1:m}\Sigma_{1:m}V_{1:m}^{\mathrm{T}}. \tag{44}$$

Here, U and V are matrices consisting of the spatial and temporal POD modes, respectively. The r-rank approximation of the observed data matrix is calculated as follows:

$$Y_{1:m} \sim \tilde{U}_{1:m}\tilde{\Sigma}_{1:m}\tilde{V}_{1:m}^{\mathrm{T}}, \tag{45}$$

where quantities with tildes indicate r-rank approximations. Here, the r-dimension matrix of $\tilde{\Sigma}$ consists of r-largest singular values of Σ. In addition, the row vectors of $\tilde{U}$ and $\tilde{V}$ are the same as the corresponding first r row vectors of U and V. Using these matrices, reduced-order $\tilde{Y}$, which represents mode strength, is constructed as follows:

$$\tilde{Y}_{1:m} = \tilde{\Sigma}_{1:m}\tilde{V}_{1:m}^{\mathrm{T}}. \tag{46}$$

In the second step (step 2), $\tilde{Y}$ and $\tilde{y}_k$ are treated in a manner similar to Y and y_k in the proposed EKFDMD procedures, and $\boldsymbol{x}_k$ and A are simultaneously estimated once. In addition, for online implementation,

$$\tilde{\boldsymbol{y}}_k = \tilde{U}^{\mathrm{T}}\boldsymbol{y} \tag{47}$$

can be used where the left singular vector is assumed to be fixed using the sample data. After this process, the eigenvalues and eigenmodes are computed by solving the eigenvalue problem of A.

Finally, in the third step (step 3), the original dimension of the eigenmode is obtained by multiplying matrix U after obtaining the right eigenvector of the reduced system by EKFDMD.

$$\boldsymbol{x}_k = \tilde{U}_{1:m}\tilde{\boldsymbol{x}}_k, \tag{48}$$

Again, note that we can use the same formulation in Eqs 46 through 48 for an online situation in which the left singular vector (spatial mode) $\tilde{U}$ is known in advance. This is similar to KFDMD [15] proposed previously. In this case, a fully online algorithm can be obtained. However, if the POD mode is not known in advance and must be estimated, then an online POD method or other methods are required. If the spatial POD modes change with time as in the case of online POD, then the projected coefficients are not consistent in time. Furthermore, the POD modes are sometimes activated or deactivated in the online POD algorithm. Thus, it appears to be difficult to straightforwardly extend the EKFDMD to a method combined with the online POD, and this is left for a future study.

In the present paper, Eq 46 is adopted for the truncated POD. This procedure is used for many-DoF problems (n>30) and is not used unless otherwise mentioned. In the case of noisy dataset, it should be noted that an accurate estimate of the mode coefficient does not necessarily mean an accurate representation of the full state because the spatial POD mode contains noise as shown later. However, despite the imperfect estimation of POD modes, eigenvalue and reconstructed data by EKFDMD are sufficiently accurate, which is also shown later.

3.3 Implementation of the EKFDMD algorithm

Here, the EKFDMD algorithm is briefly summarized. After initialization, the prediction (a priori estimation) and update steps are alternately performed.

Initialization

1. If the DoF is large, trPOD is applied to the data.
2. Set $\boldsymbol{\theta} = [\boldsymbol{y}_0^{\mathrm{T}} \quad (\mathrm{vec}(I))^{\mathrm{T}}]^{\mathrm{T}}$ and $P_{0|0} = \gamma I$. Here, γ is large. (In the present study, we set $\gamma = 1,000$).

Prediction step

1. $x_{k|k-1}$ are predicted by Eqs 36 and 34, while $a_{ij,k|k-1}$ are predicted to be the same as $a_{ij,k-1|k-1}$.
2. $P_{k|k-1}$ is predicted by Eqs 37 and 35.

Update step

1. The Kalman gain K is computed by Eqs 39 and 38.
2. $\theta_{k|k}$ is updated by Eqs 40 and 42, and matrix A is obtained using $\theta_{k|k-1}$.
3. $P_{k|k}$ is updated by Eq 43.

4 Numerical Experiments and discussion

The EKFDMD algorithm described in Section 3.1 is adopted in the numerical experiments below.

4.1 Problem with a small number of DoFs without system noise

First, the performance of EKFDMD is investigated for the standard problem, in comparison with the standard DMD, KFDMD, tlsDMD, and optDMD. The problem is approximately the same as that considered in the previous study. [13] This problem is modified slightly to involve the system noise in discretized form for the next subsection, although only the observation noise is first considered in this subsection.

The discretized eigenvalues are assumed to be positioned at $\lambda_1 = \exp[(\pm 2\pi i \Delta t)]$, $\lambda_2 = \exp[(\pm 5\pi i \Delta t)]$, and $\lambda_3 = \exp[(-0.3 \pm 11\pi i)\Delta t]$, where $\Delta t = 0.01$. The corresponding continuous eigenvalues are $\omega_1 = (\pm 2\pi i)$, $\omega_2 = (\pm 5\pi i)$, and $\omega_3 = (-0.3 \pm 11\pi i)$. The number of DoFs of this system is $d = 6$. The original data $\boldsymbol{f}$ were computed in the previous study as

$$\frac{\mathrm{d}\boldsymbol{f}}{\mathrm{d}t} = B\boldsymbol{f}, \tag{49}$$

$$B = \begin{pmatrix} |\mathrm{Re}(\omega_1)| & |\mathrm{Im}(\omega_1)| & 0 & 0 & 0 & 0 \\ -|\mathrm{Im}(\omega_1)| & |\mathrm{Re}(\omega_1)| & 0 & 0 & 0 & 0 \\ 0 & 0 & |\mathrm{Re}(\omega_2)| & |\mathrm{Im}(\omega_2)| & 0 & 0 \\ 0 & 0 & -|\mathrm{Im}(\omega_2)| & |\mathrm{Re}(\omega_2)| & 0 & 0 \\ 0 & 0 & 0 & 0 & |\mathrm{Re}(\omega_3)| & |\mathrm{Im}(\omega_3)| \\ 0 & 0 & 0 & 0 & -|\mathrm{Im}(\omega_3)| & |\mathrm{Re}(\omega_3)| \end{pmatrix}. \tag{50}$$

However, the above formulation cannot treat system noise. Therefore, the system is integrated for each time step size, and discretized system noise is added as follows:

$$\boldsymbol{f}_{k+1} = e^{B\Delta t}\boldsymbol{f}_k + \boldsymbol{v}'_k, \tag{51}$$

where $\boldsymbol{v}'$ is the system noise for the original system. Eq 51 exactly corresponds to the solution of Eq 49 for the condition in which $\boldsymbol{v}_k$ is absent. In this subsection, no system noise is considered with $\boldsymbol{v}_k = 0$.

The number of DoFs of this system is $d = 6$, which is expanded to snapshot data of $n = 16$ DoFs by applying the Q_{QR} matrix of QR decomposition of a random matrix. Note that this problem was originally extended to $n = 400$ DoF, but the number of DoFs is limited in the present study because of the computational costs of EKFDMD, as mentioned above. In this process, a random matrix T of $n \times d$ dimensions in which each of the components is a random number of $\mathcal{N}(0, 1)$ is transformed into $T = Q_{\mathrm{QR}} R_{\mathrm{QR}}$ by QR decomposition, and the original data $\boldsymbol{f}_k$ of dimension d are extended to $\boldsymbol{x}_k$ of dimension n by multiplication by a matrix Q_{QR}, as follows:

$$\boldsymbol{x}_k = Q_{\mathrm{QR}} \boldsymbol{f}_k. \tag{52}$$

Then, $\boldsymbol{y}$ data matrices are created by adding white observation noise to the original $\boldsymbol{x}$ data matrix, where the noise $\boldsymbol{w}_k$ is expressed as $\mathcal{N}(0, \sigma^2)$.

$$\boldsymbol{y}_k = \boldsymbol{x}_k + \boldsymbol{w}_k. \tag{53}$$

Here, the variance (σ_w^2) is varied as 0.0001, 0.001, 0.01, and 0.1. A total of 500 snapshots are given, and the eigenvalues of the matrix A in the final stage are analyzed.

For the initial adjustable parameters of the Kalman filter, the diagonal parts of the variance matrix are set to be 10^3. The diagonal elements of Q and R are set to be 0 and σ_w^2, respectively, and the nondiagonal elements of Q and R are set to be 0 in this subsection. The assumption of $Q = 0$ corresponds to providing the information that the system noise is absent and the system is temporally constant.

The results for the noisy data while changing the noise level are discussed. Figs 1 and 2 show the eigenvalues estimated in the representative case and in all of the 100 cases we examined by changing the random number seed, respectively. The results of the estimated eigenvalues in Figs 1 and 2 show that DMD and KFDMD do not work well for accurate estimation of the eigenvalues of the system for the case in which the noise level is high. On the other hand, tlsDMD works better than DMD and KFDMD. Furthermore, optDMD and EKFDMD appear to work the best for estimation of the eigenvalues. This might be because optDMD and EKFDMD denoises the data, and a more accurate eigenvalue of the system can be obtained by the denoised data. The system identification performance of EKFDMD appears to be better than that of tlsDMD.

The above characteristics are discussed with the quantitative data. Fig 3 shows the error of eigenvalues. The errors in the eigenvalues are defined by the norm of the closest computed eigenvalue to the true eigenvalue specified. Here, outliers were not removed in this process. The error in the eigenvalues decreases with decreasing noise strength for all methods. This plot quantitatively shows that the error basically decreases with the order of DMD as well as KFDMD, tlsDMD, EKFDMD, and optDMD. The system noise is not considered in the present problem setting, and therefore optDMD can give the best-fit curve for the all of the data points, owing to its offline procedures. On the other hand, EKFDMD incrementally updates the information and cannot use all of the data at once. Therefore, it is reasonable that optDMD works slightly better than EKFDMD.

Data reconstruction is then considered. In addition, as noted previously, EKFDMD is expected to be able to denoise the data. Fig 4, which illustrates the time-series of the true data, the observation (noisy) data and the reconstructed data of DMD, tlsDMD, KFDMD, optDMD, and EKFDMD. This plot reveals that DMD and KFDMD cannot predict the oscillation

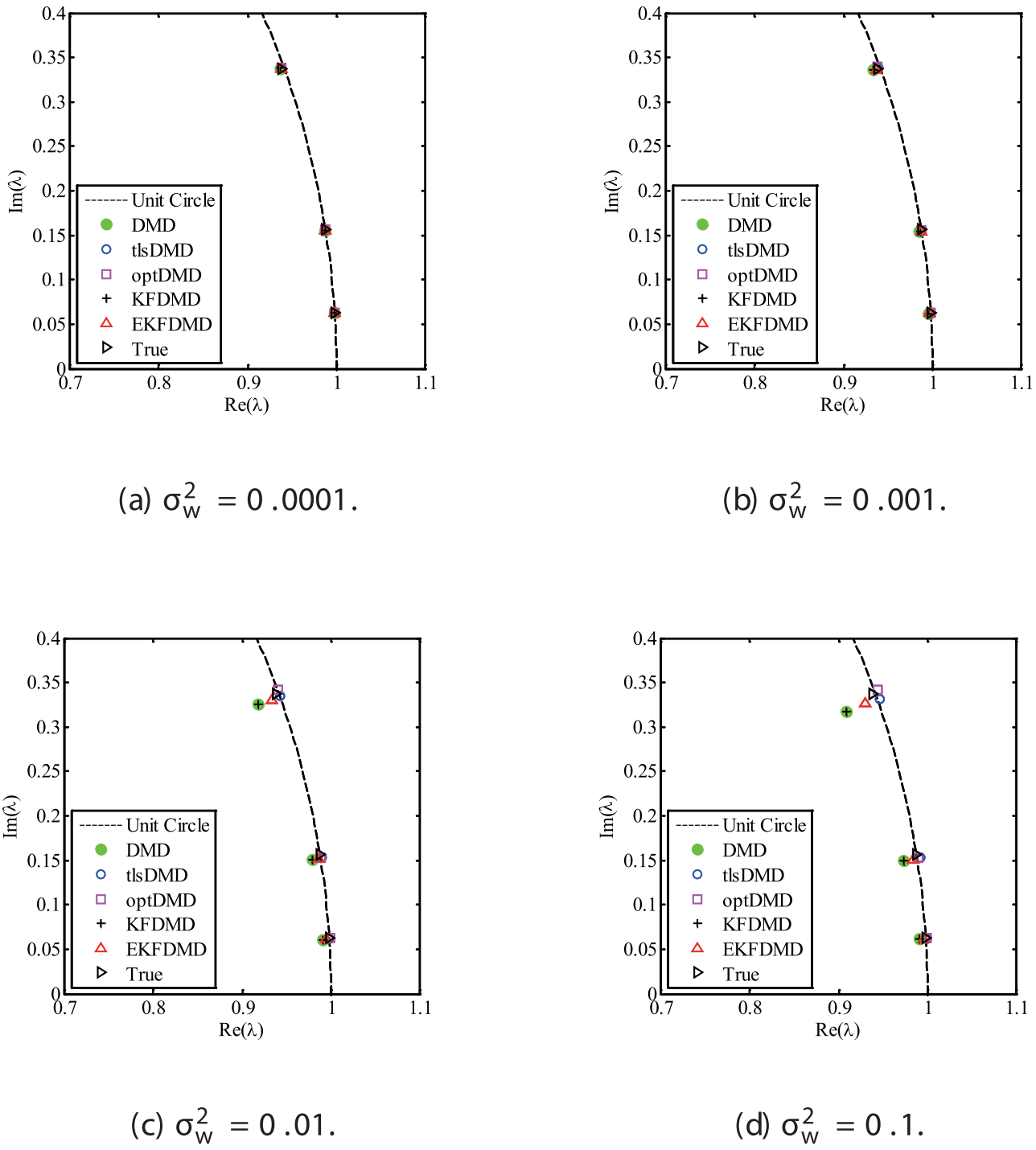
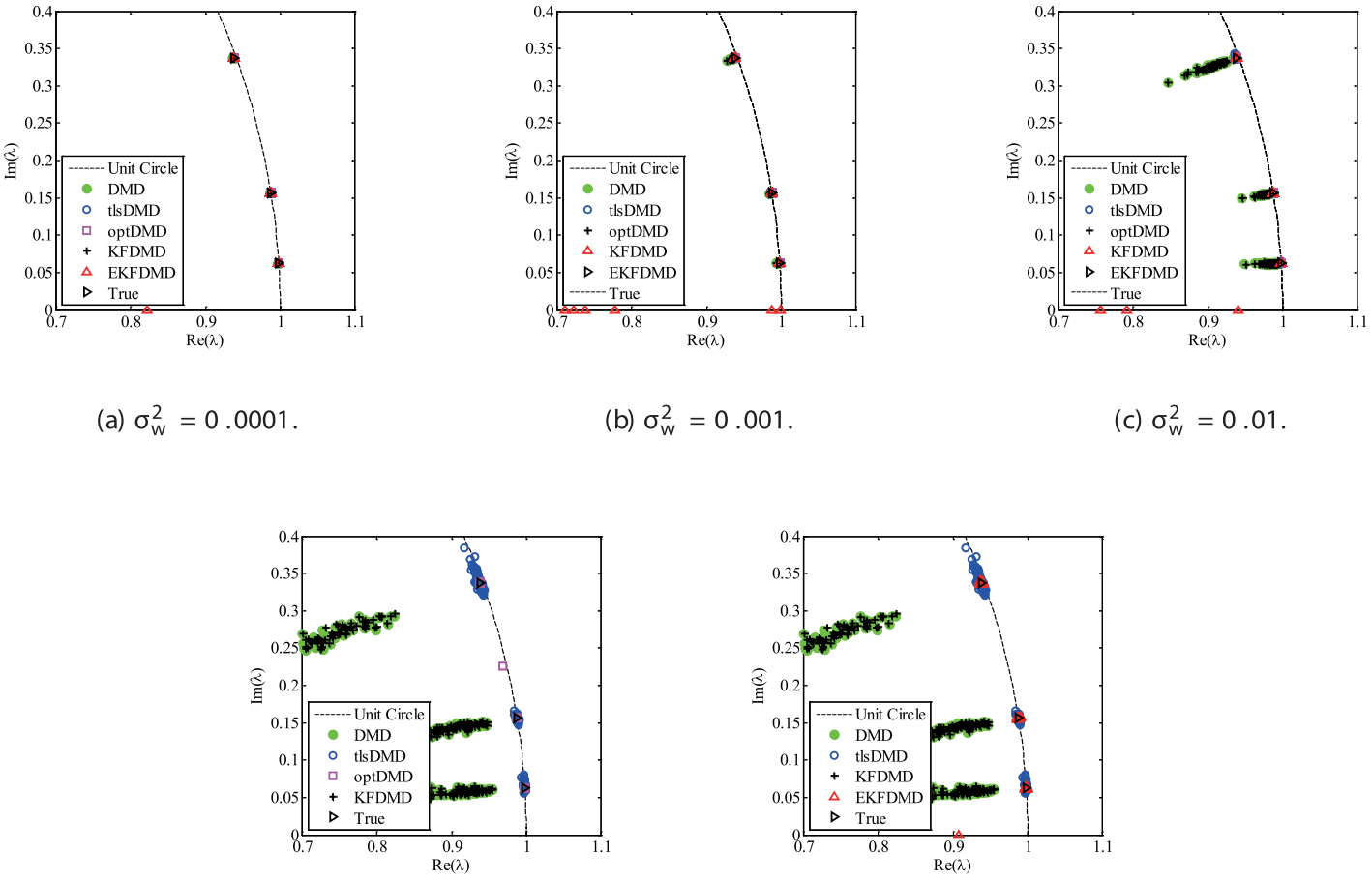


Fig 1. Eigenvalues for a problem with a small number of DoFs without system noise. The algorithms are almost identical in (a) and (b).

https://doi.org/10.1371/journal.pone.0209836.g001

because they estimate the dumping oscillation due to the noise included in the observation data. Moreover, tlsDMD can predict the oscillation for the weaker noise level. Although tlsDMD can predict neutral oscillation for a stronger noise level, as shown in Fig 4, the phase of oscillation of reconstructed data is very different from the true value. On the other hand,

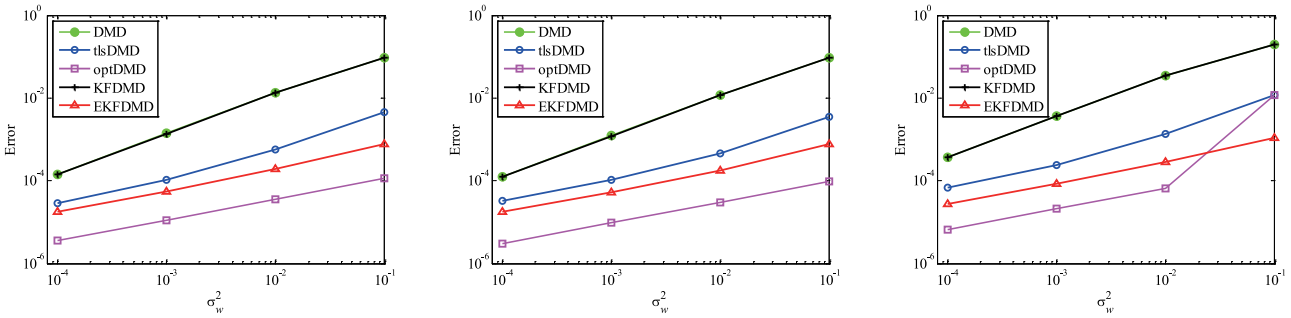


(a) $\sigma_w^2 = 0.0001$. (b) $\sigma_w^2 = 0.001$. (c) $\sigma_w^2 = 0.01$.

(d) $\sigma_w^2 = 0.1$ without EKFDMD. (e) $\sigma_w^2 = 0.1$ without optDMD.

Fig 2. Eigenvalues for multiple runs of a problem with a small number of DoFs without system noise, where the seed for the random number is different for multiple runs.

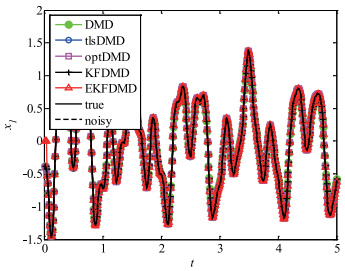
https://doi.org/10.1371/journal.pone.0209836.g002



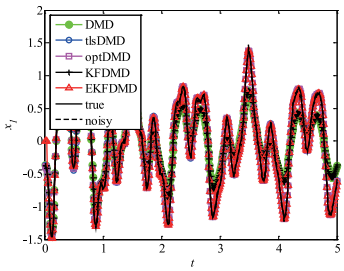
(a) λ_1. (b) λ_2. (c) λ_3.

Fig 3. Errors in the eigenvalues for multiple runs of a problem with a small number of DoFs without system noise.

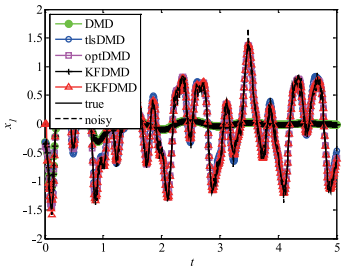
https://doi.org/10.1371/journal.pone.0209836.g003



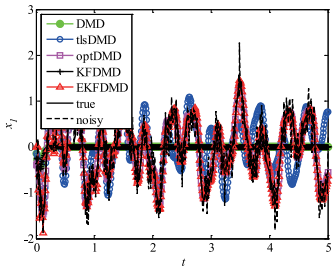
(a) $\sigma_w^2 = 0.0001$.



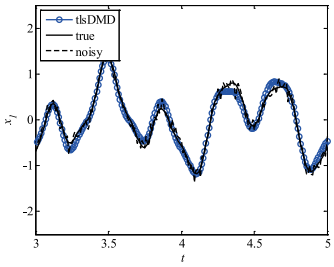
(b) $\sigma_w^2 = 0.001$.



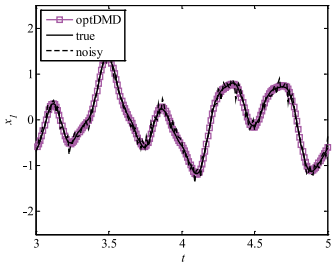
(c) $\sigma_w^2 = 0.01$.



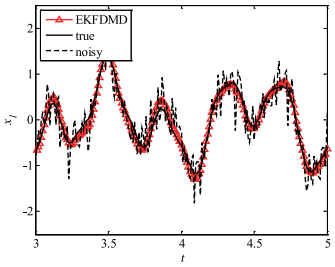
(d) $\sigma_w^2 = 0.1$.



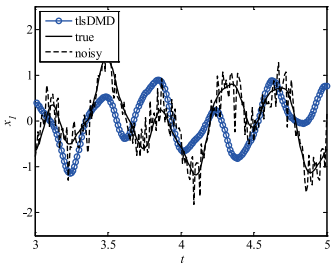
(e) $\sigma_w^2 = 0.01$, tlsDMD.



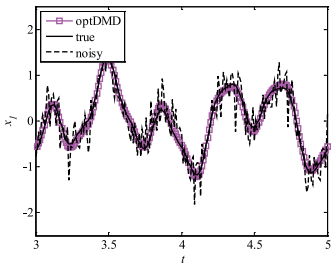
(f) $\sigma_w^2 = 0.01$, optDMD.



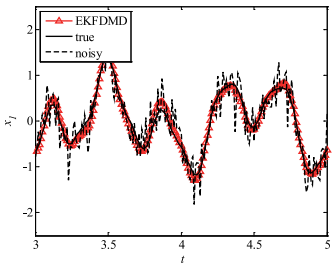
(g) $\sigma_w^2 = 0.01$, EKFDMD.



(h) $\sigma_w^2 = 0.1$, tlsDMD.



(i) $\sigma_w^2 = 0.1$, optDMD.



(j) $\sigma_w^2 = 0.1$, EKFDMD.

Fig 4. Time histories of the first node of the reconstructed data for a problem with a small number of DoFs without system noise.

https://doi.org/10.1371/journal.pone.0209836.g004

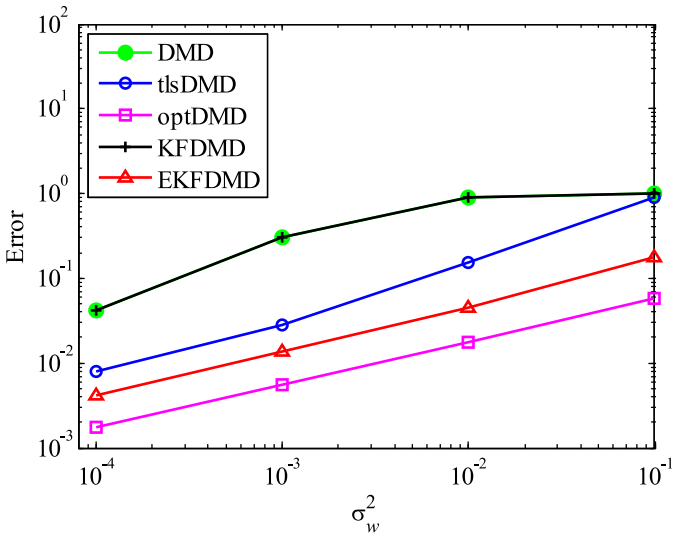


Fig 5. Errors in the reconstructed data for multiple runs of a problem with a small number of DoFs without system noise.

https://doi.org/10.1371/journal.pone.0209836.g005

optDMD and EKFDMD can successfully denoise the data, even though the noise level is very high.

The error level of the reconstructed data is quantitatively discussed in term of Fig 5, which shows the following normalized error:

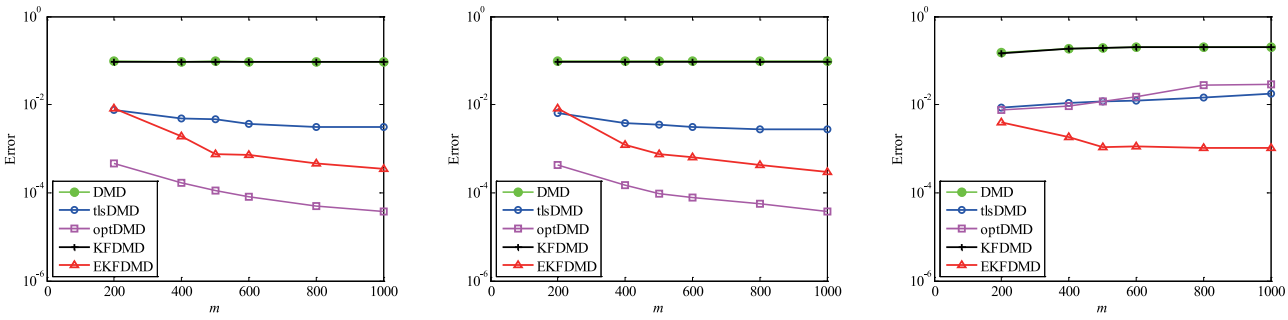
$$E_{\text{reconst}} = \frac{\| X_{\text{reconst},101:m} - X_{101:m} \|_F^2}{\| X_{101:m} \|_F^2}. \tag{54}$$

Fig 5 shows that the error decreases with the order of DMD, KFDMD, tlsDMD, EKFDMD, and optDMD, similar to those in the eigenvalues. This trend also shows that EKFDMD works reasonably for simultaneous system identification and denoising of the data by running the algorithm once. The better performance of optDMD, as compared to EKFDMD, originates from their online or offline characteristics.

Although we are interested in the performance for the case in which system noise is present, we hereinafter discuss the effects of parameters on this problem without system noise, before discussing the problem with system noise in Subsection 4.2.

4.1.1 Effect of the number of snapshots *m*. Here, the parameter effects for the problem without system noise are considered. First, the effect of the number of snapshots m is investigated. Similar to the previous discussion, the errors in the eigenvalues and reconstructed data for DMD, tlsDMD, KFDMD, optDMD, and EKFDMD are calculated for various values of m for data of $\sigma_w^2 = 0.1$. These errors are evaluated by 100 runs and are averaged for each algorithm. The error in eigenvalues in Fig 6 shows that the errors of tlsDMD, EKFDMD, and optDMD basically decrease (except for some bumps), while those of DMD and KFDMD do not. Interestingly, the error of EKFDMD decreases more rapidly and is larger than that of tlsDMD for $m \leq 200$ but smaller for $m \geq 300$. This is because EKFDMD is an one-path algorithm and its accuracy in the early stage is not sufficiently high, but increases rapidly as more successive data are obtained. Note that both tlsDMD and optDMD algorithms are offline algorithms.

Then, the errors in reconstructed data shown in Fig 7 are discussed. The errors of DMD, KFDMD, and tlsDMD do not change. The errors of DMD and KFDMD do not decrease



(a) λ_1. (b) λ_2. (c) λ_3.

Fig 6. Effect of *m* on errors in the eigenvalues for multiple runs of a problem with a small number of DoFs without system noise.

https://doi.org/10.1371/journal.pone.0209836.g006

because they cannot better predict the eigenvalues for the case in which m increases, and the errors of tlsDMD do not decrease, despite the decrease in the error in the eigenvalues, because the reconstructed data with tlsDMD have a different phase due to the very strong observation noise in the initial snapshot, as discussed previously. On the other hand, the errors of EKFDMD and optDMD decrease because both algorithms find the best-fit data for reconstruction and the accuracy of this data increases by using the information of an increased number of snapshots.

4.1.2 Effect of mismatched error level for *R*. Next, the effect of mismatched R settings is discussed, while the system error is absent and Q is set to be 0. In the present study, we investigate the mismatched cases of $R = 10\sigma_w^2 I$ and $R = 0.1\sigma_w^2 I$, as well as the matched case of $R = \sigma_w^2 I$, the results of which are presented in the previous sections. The number of snapshots m is set to be 500. The errors are evaluated by 100 runs and are averaged for each case, similar to previous cases. The errors of EKFDMD in eigenvalues and reconstructed data for the case in which R is mismatched are shown in Figs 8 and 9, respectively. These figures show that the mismatched R does not affect the results, except for the strong-observation-noise case

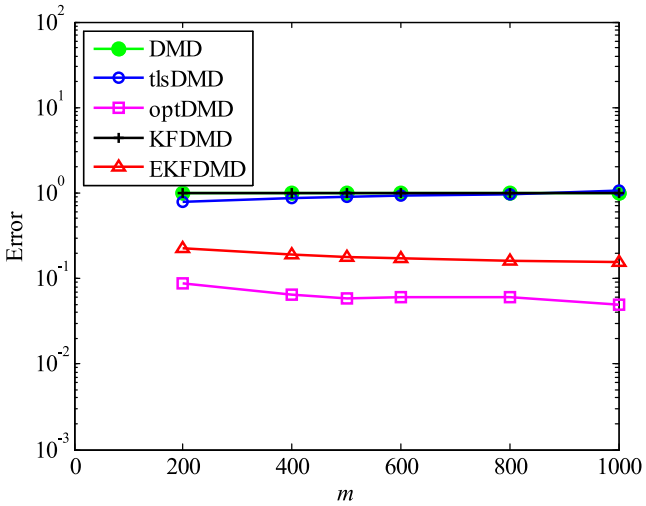
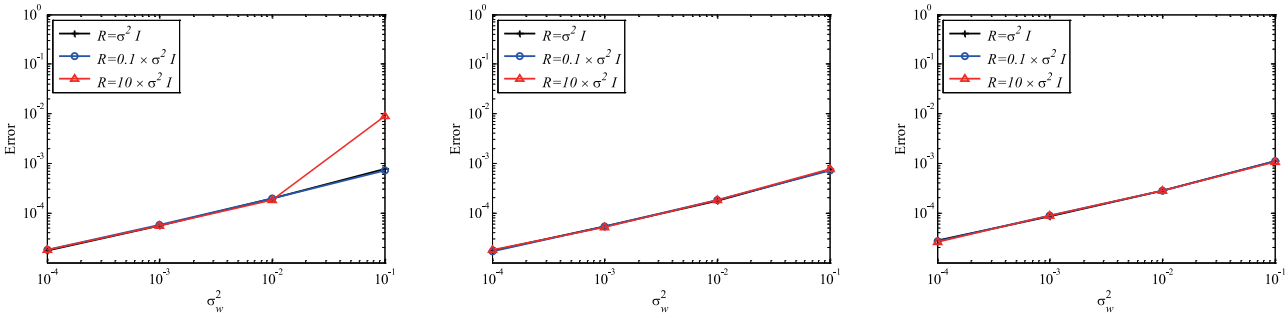


Fig 7. Effect of *m* on errors in the reconstructed data for multiple runs of a problem with a small number of DoFs without system noise.

https://doi.org/10.1371/journal.pone.0209836.g007



(a) λ_1. (b) λ_2. (c) λ_3.

Fig 8. Effect of *R* on errors in the eigenvalues for multiple runs of a problem with a small number of DoFs without system noise.

https://doi.org/10.1371/journal.pone.0209836.g008

($\sigma_w^2 = 0.1$), because the balance of R and Q changes the behavior of Kalman filter, whereas a change in R under the condition of $Q = 0$ does not affect the behavior of Kalman filter.

4.2 Problem with a small number of DoFs with system noise

Next, we consider a problem with system noise. In this problem, $\boldsymbol{\nu}'$ is assumed to be $\mathcal{N}(0, n\sigma_v^2/6)$, resulting in $\boldsymbol{\nu}$ being $\mathcal{N}(0, \sigma_v^2)$, and we vary $\sigma_v^2 = \sigma_w^2$ as 0.0001, 0.001, 0.01, and 0.1. A hyperparameter Q is set to be

$$Q = \begin{bmatrix} Q_{1,1} & Q_{1,2} \\ Q_{2,1} & Q_{2,2} \end{bmatrix} = \begin{bmatrix} \sigma_v^2 I_{n\times n} & 0 \\ 0 & 0 \end{bmatrix} \tag{55}$$

and R is set to be $\sigma_w^2 I$. The number of snapshots m is set to be 500, and a total of 100 runs are conducted for each case.

Figs 10 and 11 show the eigenvalues estimated in the representative case and in all 100 cases we examined by changing the seed of the random numbers, respectively. Figs 10 and 11 show

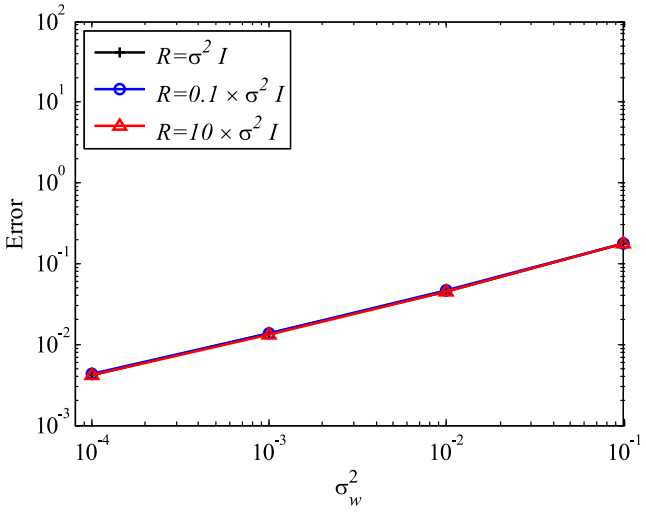


Fig 9. Effect of *R* on errors in reconstructed data for multiple runs of a problem with a small number of DoFs without system noise.

https://doi.org/10.1371/journal.pone.0209836.g009

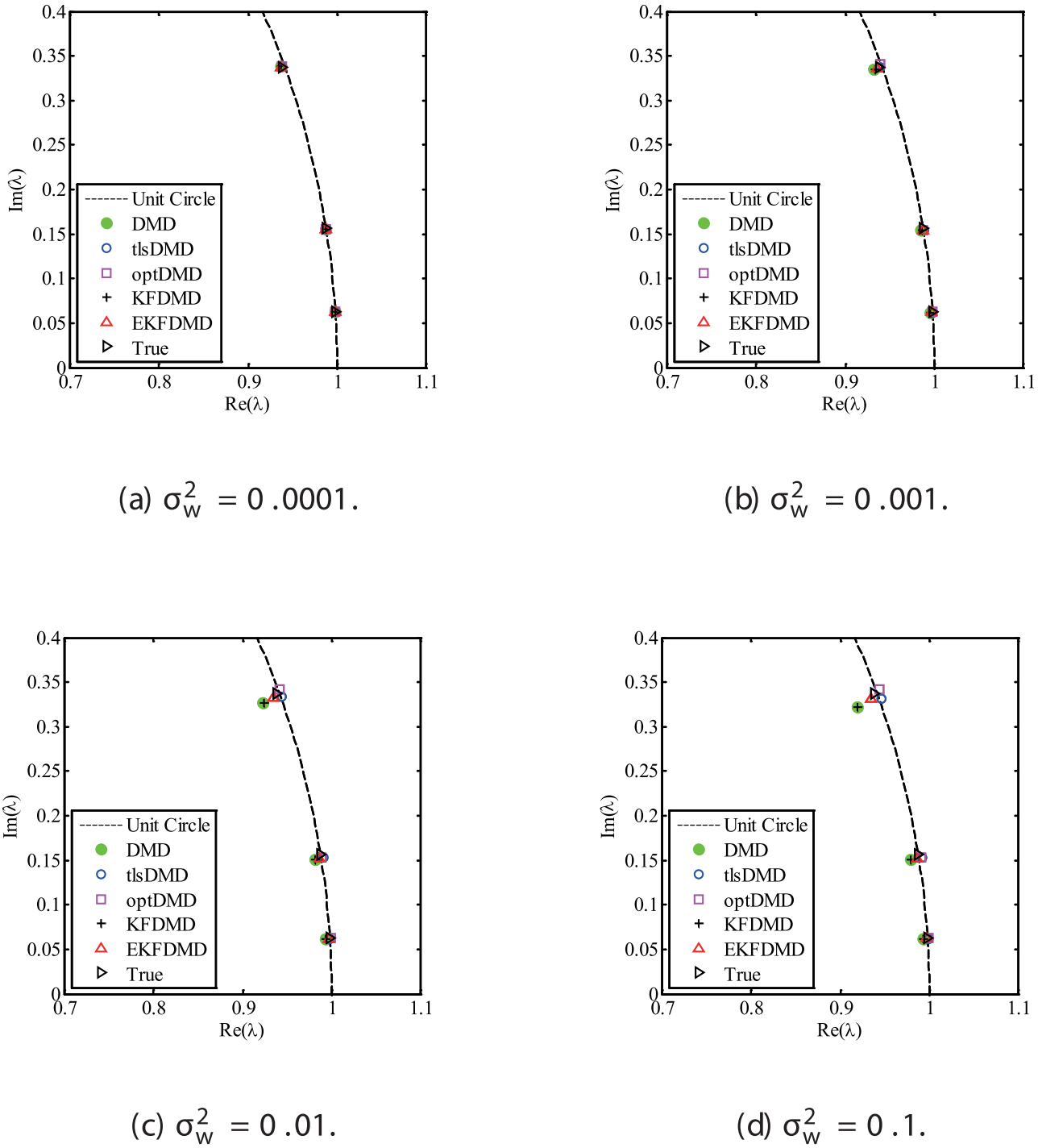
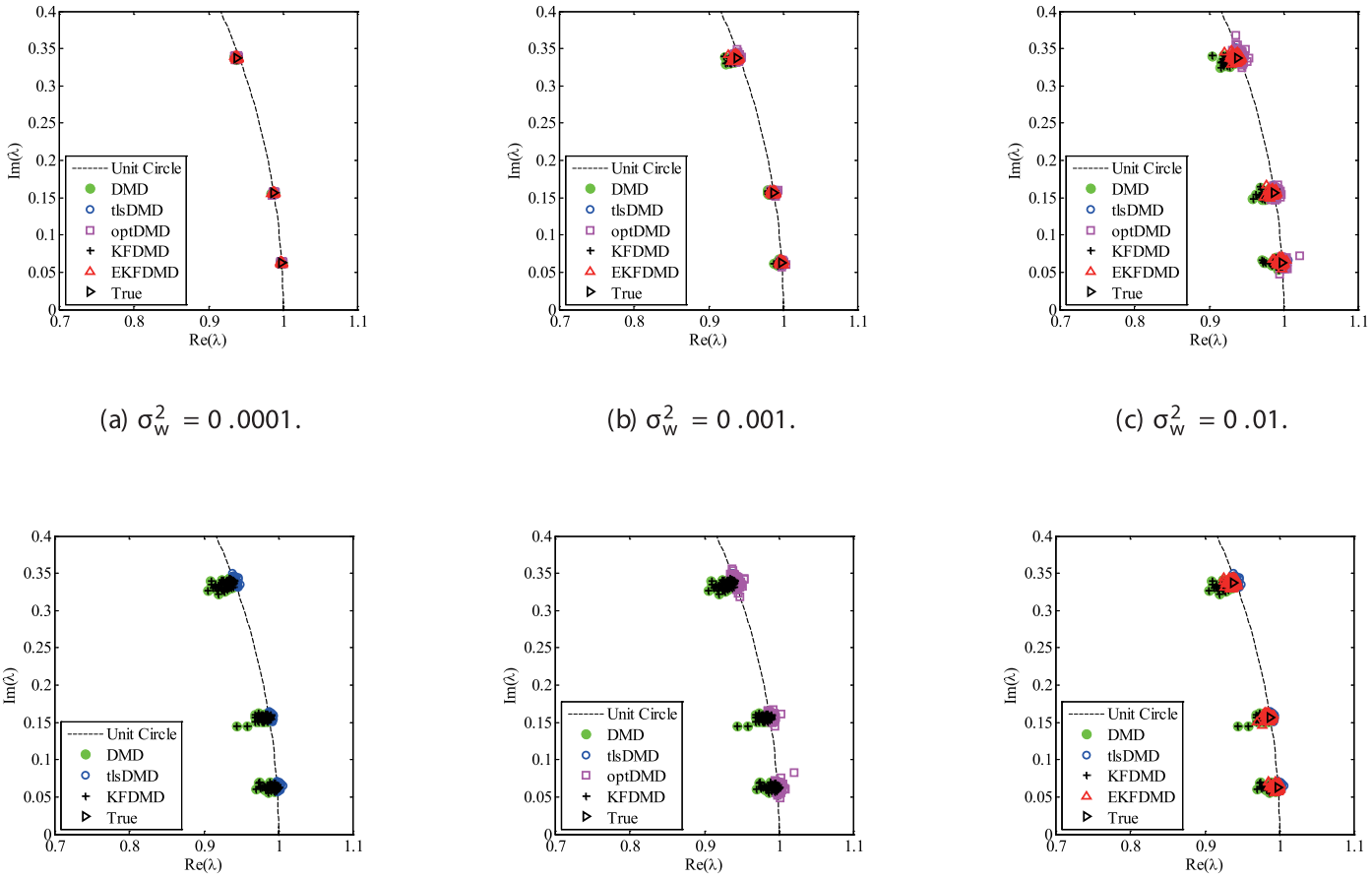


Fig 10. Eigenvalues for a problem with a small number of DoFs with system noise. The algorithms are almost identical in (a) and (b), and tlsDMD, optDMD, and EKFDMD are almost identical in (c) and (d).

https://doi.org/10.1371/journal.pone.0209836.g010

that DMD and KFDMD do not work well for the accurate estimation of the eigenvalues of the system for the case in which the noise level is high, although its accuracy is somehow improved compared with the case without the system noise. On the other hand, tlsDMD, optDMD and EKFDMD appear to work better than DMD or KFDMD. This might be because denoising



(a) $\sigma_w^2 = 0.0001$. (b) $\sigma_w^2 = 0.001$. (c) $\sigma_w^2 = 0.01$.

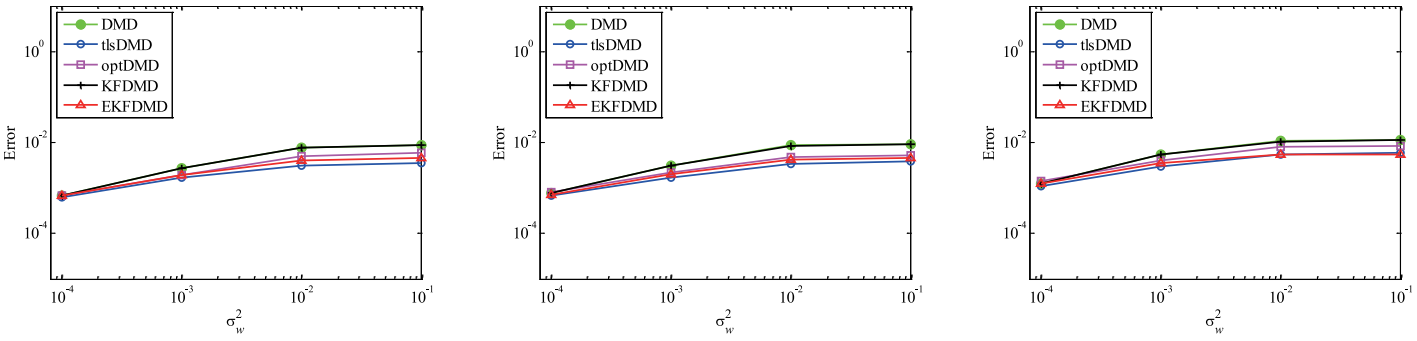
(d) $\sigma_w^2 = 0.1$ without optDMD and EKFDMD. (e) $\sigma_w^2 = 0.1$ without EKFDMD. (f) $\sigma_w^2 = 0.1$ without optDMD.

Fig 11. Eigenvalues for multiple runs of a problem with a small number of DoFs with system noise, where the seed for the random number is different for multiple runs.

https://doi.org/10.1371/journal.pone.0209836.g011

algorithms for estimation of eigenvalues of tlsDMD, optDMD, and EKFDMD works well for these data, and a more accurate eigenvalue of the system can be obtained. The system identification performance of EKFDMD appears to be as good as that of tlsDMD and optDMD in these plots. Finally, the errors of eigenvalue estimation are shown in Fig 12. Fig 12 shows that tlsDMD, optDMD, and EKFDMD work better than DMD and KFDMD. Among tlsDMD, optDMD, and EKFDMD, tlsDMD works slightly better for λ_1 and λ_2, whereas the performance of EKFDMD is similar to that of tlsDMD for λ_3. This result illustrates that the system identification performances of tlsDMD, optDMD, and EKFDMD are approximately the same for the case in which system noise is present.

Then, reconstruction using these algorithms, as shown in Fig 13, is discussed. Similar to the cases without system noise, data reconstructed by DMD and KFDMD are dumped in the early stage. This is again because the these algorithms predict dumping modes. The data reconstructed by tlsDMD have good amplitude of oscillations, but their phases do not match well with those of the original data. Although the data reconstructed by optDMD have good amplitude and phase, the data around peaks are sometimes not reconstructed. These errors around



(a) λ_1. (b) λ_2. (c) λ_3.

Fig 12. Errors in the eigenvalues for multiple runs of a problem with a small number of DoFs without system noise.

https://doi.org/10.1371/journal.pone.0209836.g012

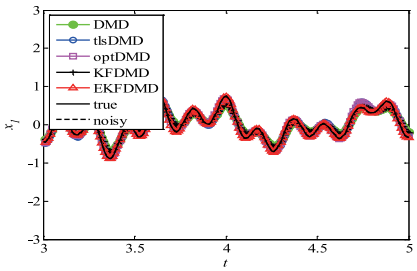
peaks in the reconstruction data obtained using optDMD are caused by system noise in the data because optDMD cannot handle system noise. Unlike the algorithm described above, the data reconstructed by EKFDMD shows excellent agreement with the original data. This is because EKFDMD can handle data with system noise. This characteristic can be used for simultaneous system identification and denoising of data containing system noise. The error in the reconstructed data shown in Fig 14 clearly shows this characteristic.

4.2.1 Effects of the balance of system and observation noises. In this subsubsection, the effects of the balance of system and observation noises in the observation data are discussed. System noise variance σ_v^2 is set to be $10\sigma_w^2$ and $0.1\sigma_w^2$. Here, Q and R are correctly given in this problem. In both cases, test cases with σ_w^2 of 0.0001, 0.001, 0.01, and 0.1 are conducted, and the results of 100 runs with different seeds for random numbers are averaged for error characteristics.

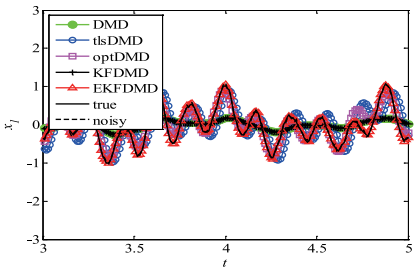
First, the case with strong system noise $\sigma_v^2 = 10\sigma_w^2$ is discussed. The errors in the estimated eigenvalues shown in Fig 15 indicate that the errors of all of the algorithms are almost the same and the error does not decrease with decreasing noise level. This figure shows that advanced DMD methods do not significantly improve the estimation of eigenvalues for data with strong system noise. The error in the reconstructed data is shown in Fig 16. This figure shows that the error of EKFDMD is much less than the errors of the other algorithms. This indicates that EKFDMD can be used for noise reduction for the case in which the system noise is stronger than the observation noise.

Then, the case with the weaker system noise $\sigma_v^2 = 0.1\sigma_w^2$ is discussed. Again, Q and R are correctly given in this problem. The error plots in Fig 17 show that the errors of tlsDMD, optDMD, and EKFDMD are approximately the same and are lower than those of DMD and KFDMD. This figure illustrates that advanced DMD methods improve the estimation ability of eigenvalues. The error in the reconstructed data is shown in Fig 18. This plot indicates that the errors decrease in the order of DMD and KFDMD (same as that of DMD), tlsDMD, optDMD, and EKFDMD. The figure also shows that EKFDMD performs better than optDMD, even if weaker system noise is present. This fact indicates that EKFDMD can be used for noise reduction in the range we investigated for the case in which system noise is present, regardless of its strength.

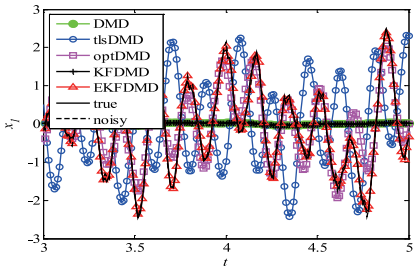
4.2.2 Effects of mismatched error level for *Q* and *R*. In this subsubsection, the effects of mismatched selection of Q and R are discussed. The system noise variance σ_v^2 is set to be the same as σ_w^2. First, the effect of mismatched Q is discussed. Fig 19 shows that mismatched Q



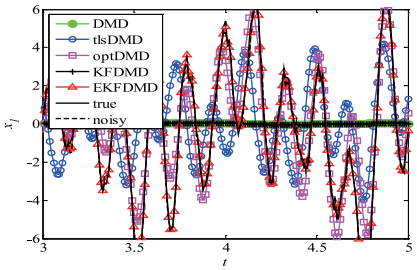
(a) $\sigma_w^2 = 0.0001$.



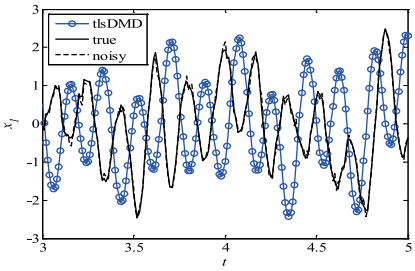
(b) $\sigma_w^2 = 0.001$.



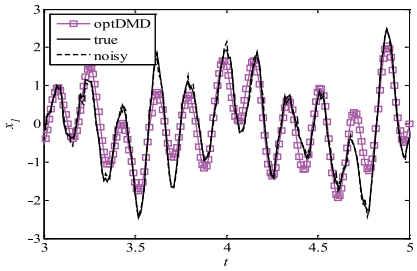
(c) $\sigma_w^2 = 0.01$.



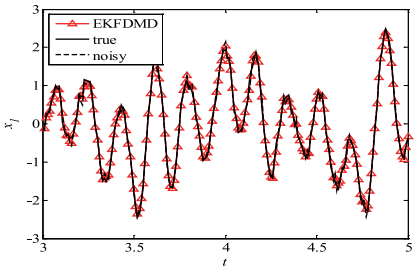
(d) $\sigma_w^2 = 0.1$.



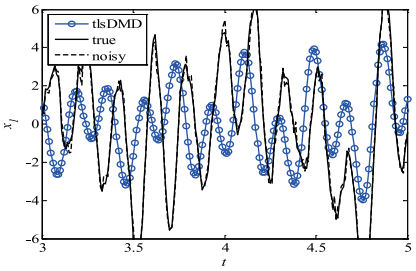
(e) $\sigma_w^2 = 0.01$, tlsDMD.



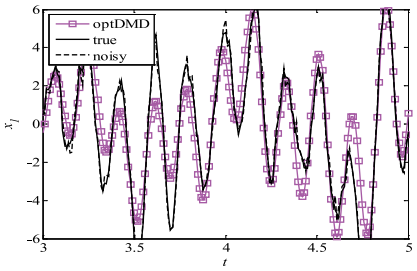
(f) $\sigma_w^2 = 0.01$, optDMD.



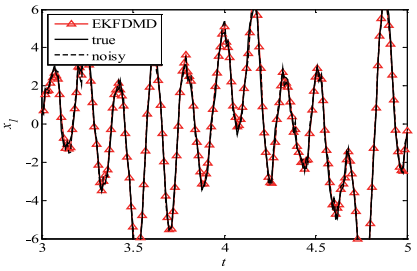
(g) $\sigma_w^2 = 0.01$, EKFDMD.



(h) $\sigma_w^2 = 0.1$, tlsDMD.



(i) $\sigma_w^2 = 0.1$, optDMD.



(j) $\sigma_w^2 = 0.1$, EKFDMD.

Fig 13. Reconstructed data of the first node for a problem with a small number of DoFs with system noise.

https://doi.org/10.1371/journal.pone.0209836.g013

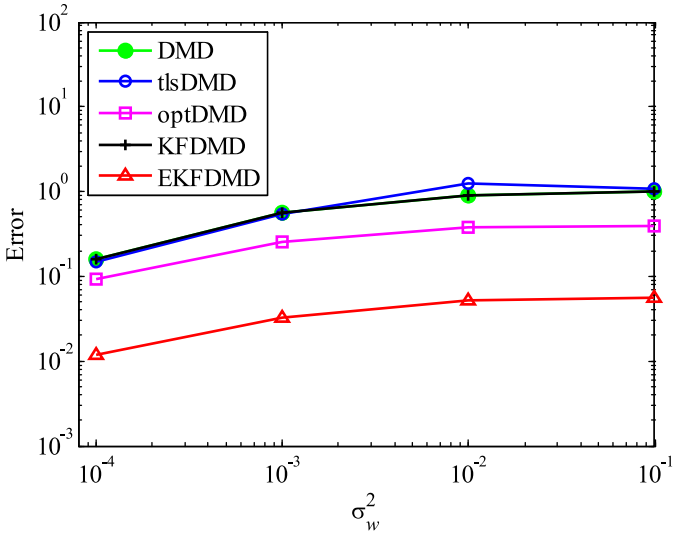
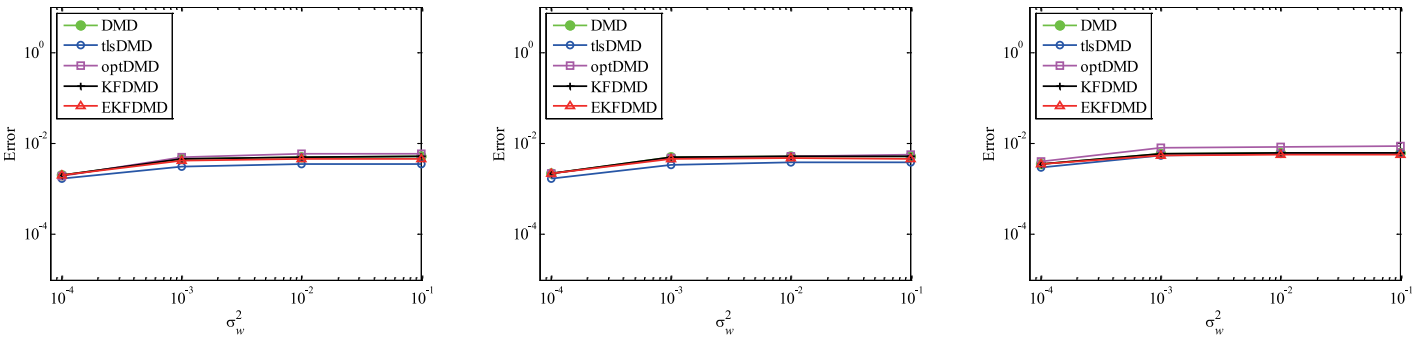


Fig 14. Errors in the reconstructed data for multiple runs of a problem with a small number of DoFs with system noise.

https://doi.org/10.1371/journal.pone.0209836.g014

does not significantly affect the error in the estimated eigenvalues, although the result with the appropriate setting (matched Q of $Q_{1,1} = \sigma_w^2 I$) exhibits the best performance. Fig 20 shows the errors in reconstructed data with the mismatched Q. In this case, if Q is assumed to be zero, which corresponds to the assumption of no system noise, then the error becomes noticeably larger. On the other hand, if Q is set to be 10 times or 0.1 times larger than the appropriate value, then the results are not significantly degraded. This indicates that the setting of Q does not significantly affect the results if the system noise is considered and Q is appropriately set to be within the order of σ_v^2.

Then, the effect of mismatched R is discussed. The error in estimated eigenvalues shown in Fig 21 illustrates that the mismatched R does not significantly change the error, although errors for smaller R or $R = 0$ become slightly larger. Fig 22 shows the errors in reconstructed data with mismatched R. In this case, mismatched R does not significantly affects the results.



(a) λ_1. (b) λ_2. (c) λ_3.

Fig 15. Errors in the eigenvalues for multiple runs of a problem with a small number of DoFs without system noise for the case in which $\sigma_v^2 = 10\sigma_w^2$.

https://doi.org/10.1371/journal.pone.0209836.g015

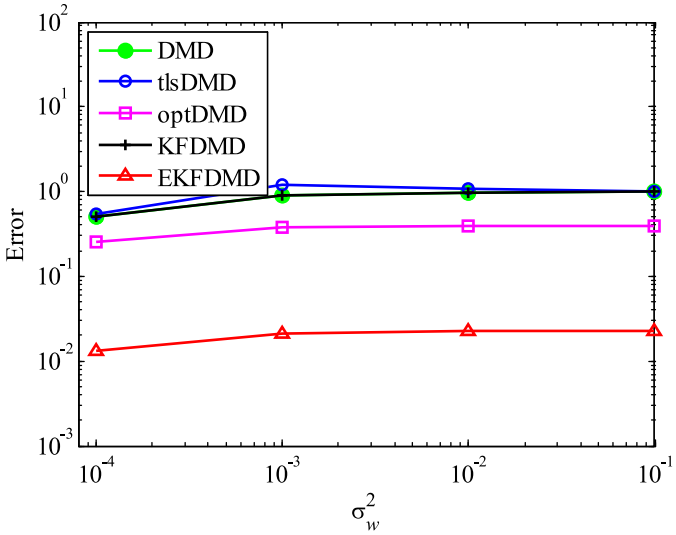


Fig 16. Errors in the reconstructed data for multiple runs of a problem with a small number of DoFs with system noise for the case in which $\sigma_v^2 = 10\sigma_w^2$.

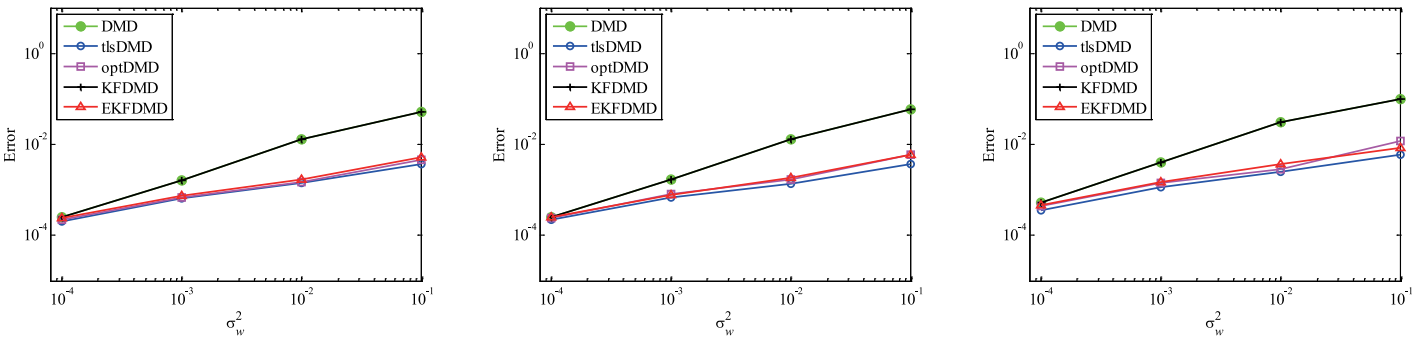
https://doi.org/10.1371/journal.pone.0209836.g016

This result shows that the setting of R does not significantly affect the results, similar to the mismatched Q cases.

Finally, the effects of mismatched Q and R, but with the condition $Q_{1,1} = R$, are investigated. The errors in the estimated eigenvalues and reconstructed data for the cases in which $Q_{1,1} = R = 10\sigma_w^2 I = 10\sigma_v^2 I$ and $Q_{1,1} = R = 0.1\sigma_w^2 I = 0.1\sigma_v^2 I$ are shown in Figs 23 and 24, respectively. These figures show that the results do not change for the case in which the ratio of $Q_{1,1}$ to R does not change. As noted earlier, the ratio of Q and R should be carefully chosen in order to achieve accurate estimation.

4.3 Problem with a moderate number of DoFs without system noise

Next, a similar problem, but with the number of DoFs extended to 200 by the same procedure, is adopted with the same noise levels. In this case, the computational cost is very high, and we



(a) λ_1. (b) λ_2. (c) λ_3.

Fig 17. Errors in the eigenvalues for multiple runs of a problem with a small number of DoFs with system noise for the case in which $\sigma_v^2 = 0.1\sigma_w^2$.

https://doi.org/10.1371/journal.pone.0209836.g017

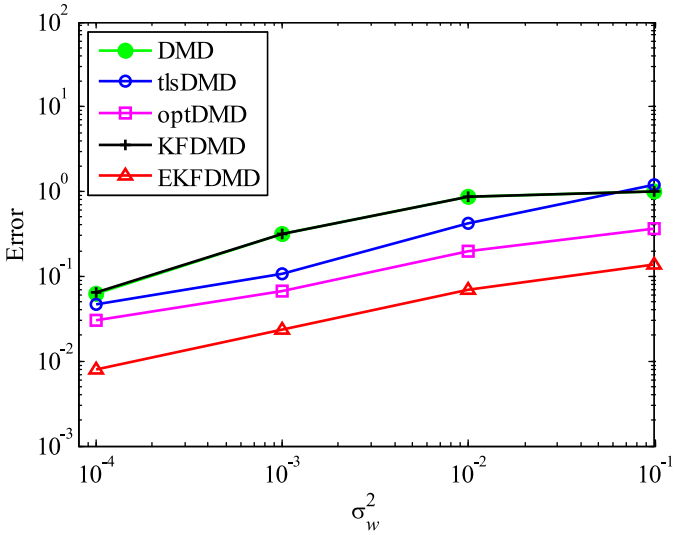
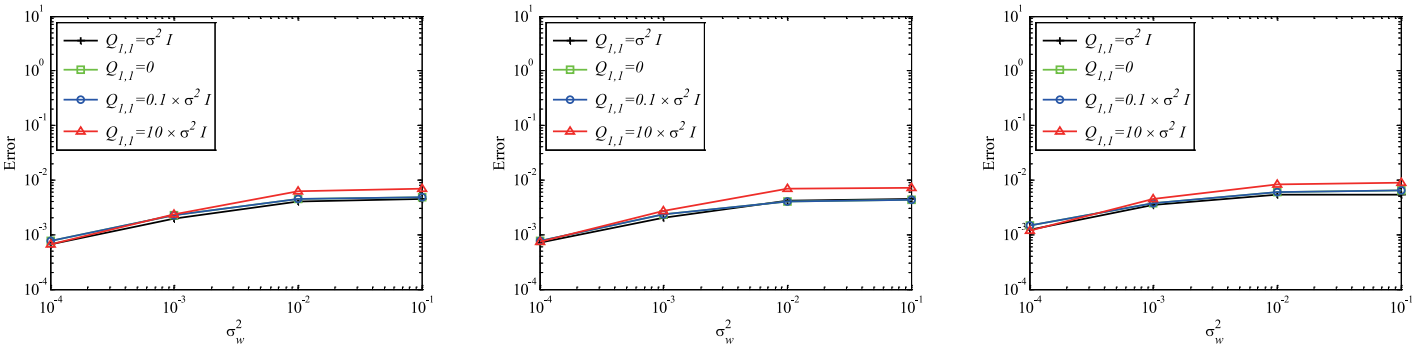


Fig 18. Errors in the reconstructed data for multiple runs of a problem with a small number of DoFs with system noise for the case in which $\sigma_v^2 = 0.1\sigma_w^2$.

https://doi.org/10.1371/journal.pone.0209836.g018

conducted trPOD as a preconditioner. In this problem, first, the number of DoFs is reduced from 200 to 10 by trPOD, and the reduced data are processed by EKFDMD. On the other hand, for the purpose of comparison, DMD and tlsDMD are applied directly to the data for 200 DoFs in order to reduce the number of DoFs to 10 because these algorithms can treat a data matrix of this size within a reasonable computational time by inherently involving truncated SVD (same as trPOD). In this problem, 500 samples were given. Similar to the previous example, the diagonal elements of the covariance matrix were set to be 10^3 in the initial condition. The diagonal elements of Q and R are set to be 0 and σ_w^2, respectively, and their nondiagonal elements are set to be 0.

The results of trPOD are shown in Fig 25, where the first POD spatial mode obtained by data without noise and that obtained by data with noise are plotted together. Note that the mode of the node distribution in snapshots is referred to as the POD spatial mode, which is analogous to fluid analysis. This plot indicates that the noise level is very high and that the



(a) λ_1. (b) λ_2. (c) λ_3.

Fig 19. Effect of mismatched Q on the errors in the eigenvalues for multiple runs of a problem with a small number of DoFs with system noise.

https://doi.org/10.1371/journal.pone.0209836.g019

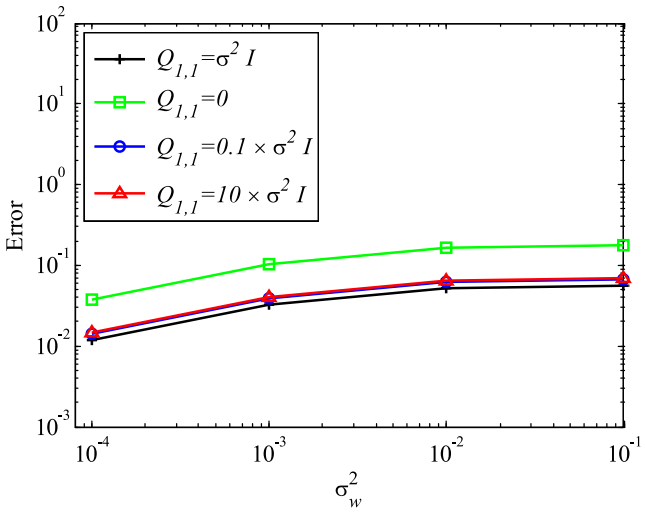
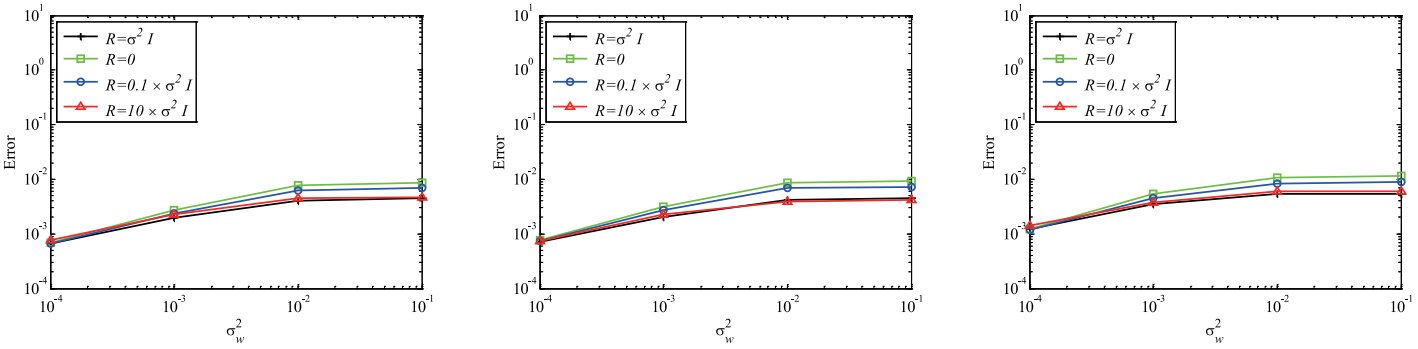


Fig 20. Effect of mismatched *Q* on the errors in the reconstructed data for multiple runs of a problem with a small number of DoFs with system noise.

https://doi.org/10.1371/journal.pone.0209836.g020

estimation of the POD spatial mode is not accurate. However, the contaminated POD modes obtained by data with noise are used for EKFDMD.

The eigenvalues and their errors for this problem are shown in Figs 26, 27, and 28. Except for the condition with strong noise ($\sigma_w^2 = 0.1$), trPOD+EKFDMD works better than DMD, KFDMD, and tlsDMD, while optDMD works best. This characteristic does not change from the small-degree-of-freedom problem, as shown earlier. The degradation in performance of the trPOD+EKFDMD for the very noisy condition might occur because the important signal is filtered out in the POD procedure. This characteristic is relaxed by increasing the number of POD modes, as shown later herein, but the number of POD modes is in a trade-off relationship with the computational cost. The reconstructed data are then shown in Fig 29. Even if we apply POD, the reconstructed data of trPOD+EKFDMD and optDMD agree well with the original data in all the condition, whereas DMD, KFDMD, and tlsDMD fail to capture the behavior of the original data in the severe noise cases. The error in the reconstructed data is shown in Fig 30. As shown earlier, the error of trPOD+EKFDMD is smaller than that of



(a) λ_1. (b) λ_2. (c) λ_3.

Fig 21. Effect of mismatched *R* on the errors in the eigenvalues for multiple runs of a problem with a small number of DoFs with system noise.

https://doi.org/10.1371/journal.pone.0209836.g021

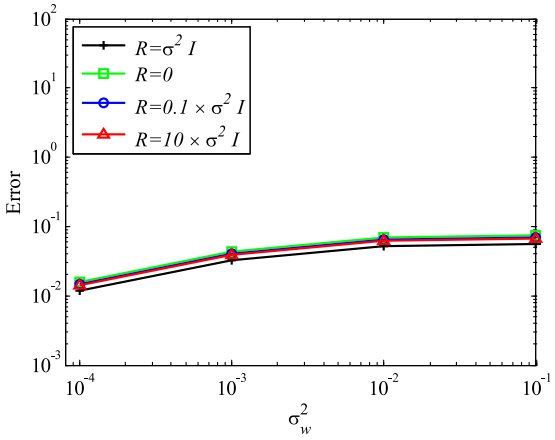
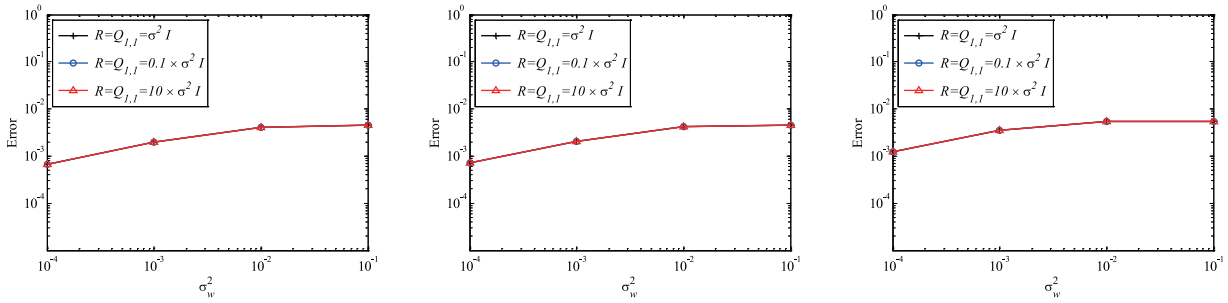


Fig 22. Effect of mismatched *R* on the errors in the reconstructed data for multiple runs of a problem with a small number of DoFs with system noise.

https://doi.org/10.1371/journal.pone.0209836.g022



(a) λ_1. (b) λ_2. (c) λ_3.

Fig 23. Effects of mismatched *Q* and *R* on the errors in the reconstructed data for multiple runs of a problem with a small number of DoFs with system noise.

https://doi.org/10.1371/journal.pone.0209836.g023

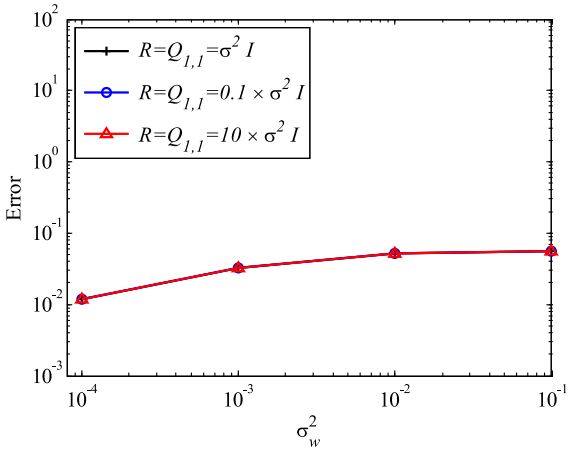


Fig 24. Effects of mismatched *Q* and *R* on the errors in the reconstructed data for multiple runs of a problem with a small number of DoFs with system noise.

https://doi.org/10.1371/journal.pone.0209836.g024

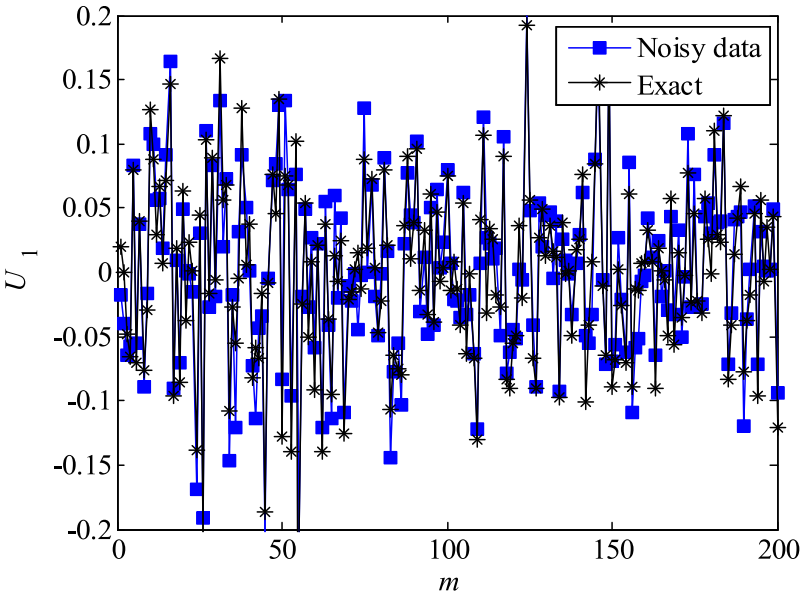


Fig 25. First POD mode of original and noisy data for a problem with a moderate number of DoFs. Here, the first POD modes of the most noisy case ($\sigma_w^2 = 0.1$) are shown.

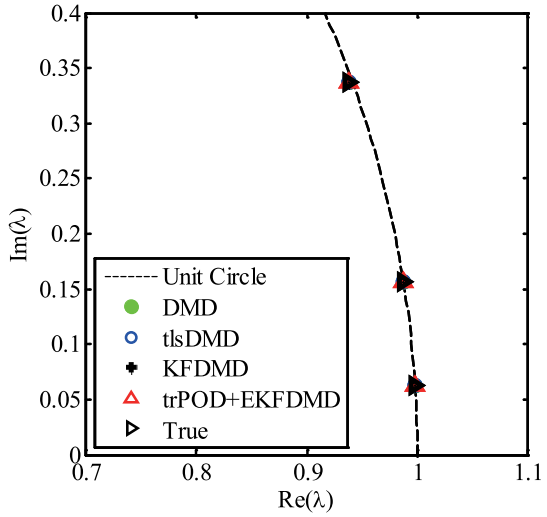
https://doi.org/10.1371/journal.pone.0209836.g025

tlsDMD and is larger than that of optDMD. Thus, trPOD+EKFDMD works reasonably well in reconstructing the data even with the imperfect POD modes shown in Fig 25.

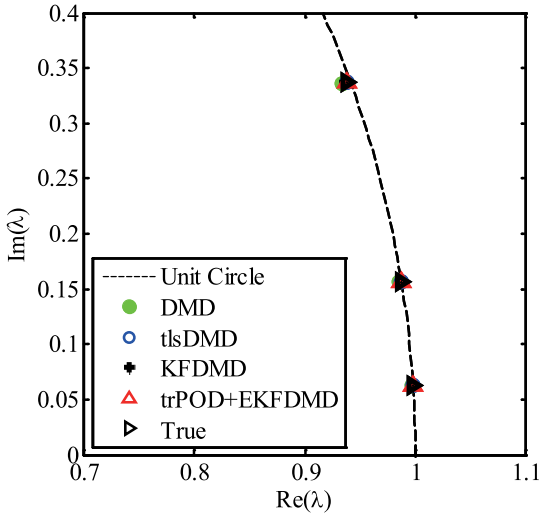
4.3.1 Effect of POD truncation. For POD truncation, the rank number should be manually specified. Therefore, the effect of the rank number chosen by the user is investigated. Here, $r = 6$ and $r = 20$ are investigated, where the previous standard cases were computed with $r = 10$, as noted earlier. The errors in the estimated eigenvalues and reconstructed data of the $r = 6$ and $r = 20$ conditions are shown in Figs 31, 32, 33 and 34, respectively. For the case in which system noise is absent, the errors of the estimation of eigenvalues by trPOD+EKFDMD does not work well with $r = 6$ for $\sigma_w^2 \geq 0.01$, and the resulting error in reconstructed data is slightly worse than that for tlsDMD for all cases with different noise levels. This might be because trPOD filters out the important signal and trPOD+EKFDMD cannot recover the original signal for strong-noise cases. On the other hand, the errors in the estimated eigenvalues of trPOD+EKFDMD with the $r = 20$ setting are lower than those of tlsDMD or are approximately the same as (and sometimes slightly higher than) that of tlsDMD and the error in the reconstructed data of trPOD+EKFDMD with $r = 20$ is smaller than that of tlsDMD. Therefore, using tnPOD+EKFDMD with better performance requires a larger rank. This is clear trade-off between the estimation accuracy and the computational cost.

4.4 Problem with a moderate number of DoFs with system noise

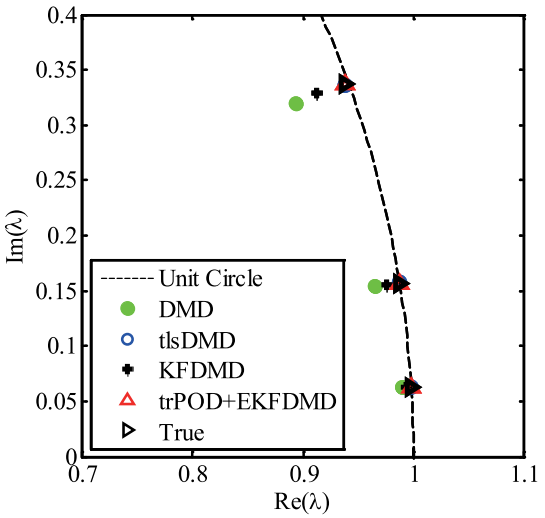
Next, we consider a similar problem in which system noise is adopted. The system noise variance σ_v^2 is set to be σ_w^2, similar to the small-DoF problem shown earlier. With regard to the EKFDMD procedure, trPOD is used as a preconditioner similar to the previous subsection. Again, in this problem, the number of DoFs is reduced from 200 to 10 by trPOD, and the reduced data are processed by EKFDMD. On the other hand, DMD, tlsDMD, and optDMD are applied directly to the data for 200 DoFs in order to reduce the number of DoFs to 10. Moreover, in this problem, 500 samples were given. The diagonal elements of the covariance



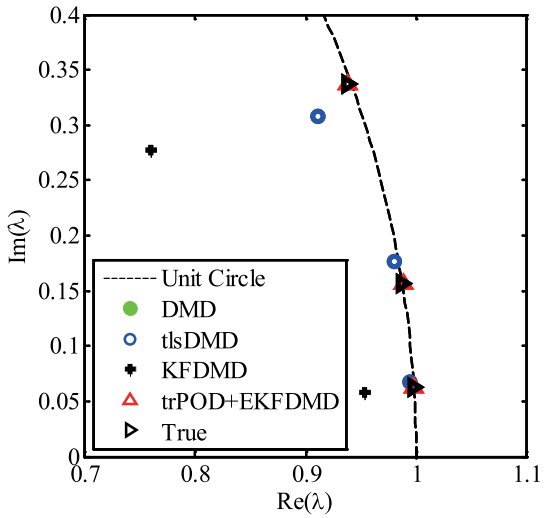
(a) $\sigma_w^2 = 0.0001$.



(b) $\sigma_w^2 = 0.001$.



(c) $\sigma_w^2 = 0.01$.



(d) $\sigma_w^2 = 0.1$.

Fig 26. Eigenvalues for a problem with a small number of DoFs without system noise. Here, rank r is set to be 10. The algorithms are almost identical in (a) and (b), and optDMD, and trPOD+EKFDMD are almost identical in (c) and (d).

https://doi.org/10.1371/journal.pone.0209836.g026

matrix are set to be 10^3 in the initial condition. The diagonal elements of R and $Q_{1,1}$ are set to be σ_w^2 and σ_v^2, respectively, and the nondiagonal elements of R and $Q_{1,1}$ are set to be 0.

The eigenvalues and their errors for this problem are shown in Figs 35, 36 and 37. Interestingly, all the algorithm work similarly each other in this condition. The degradation in

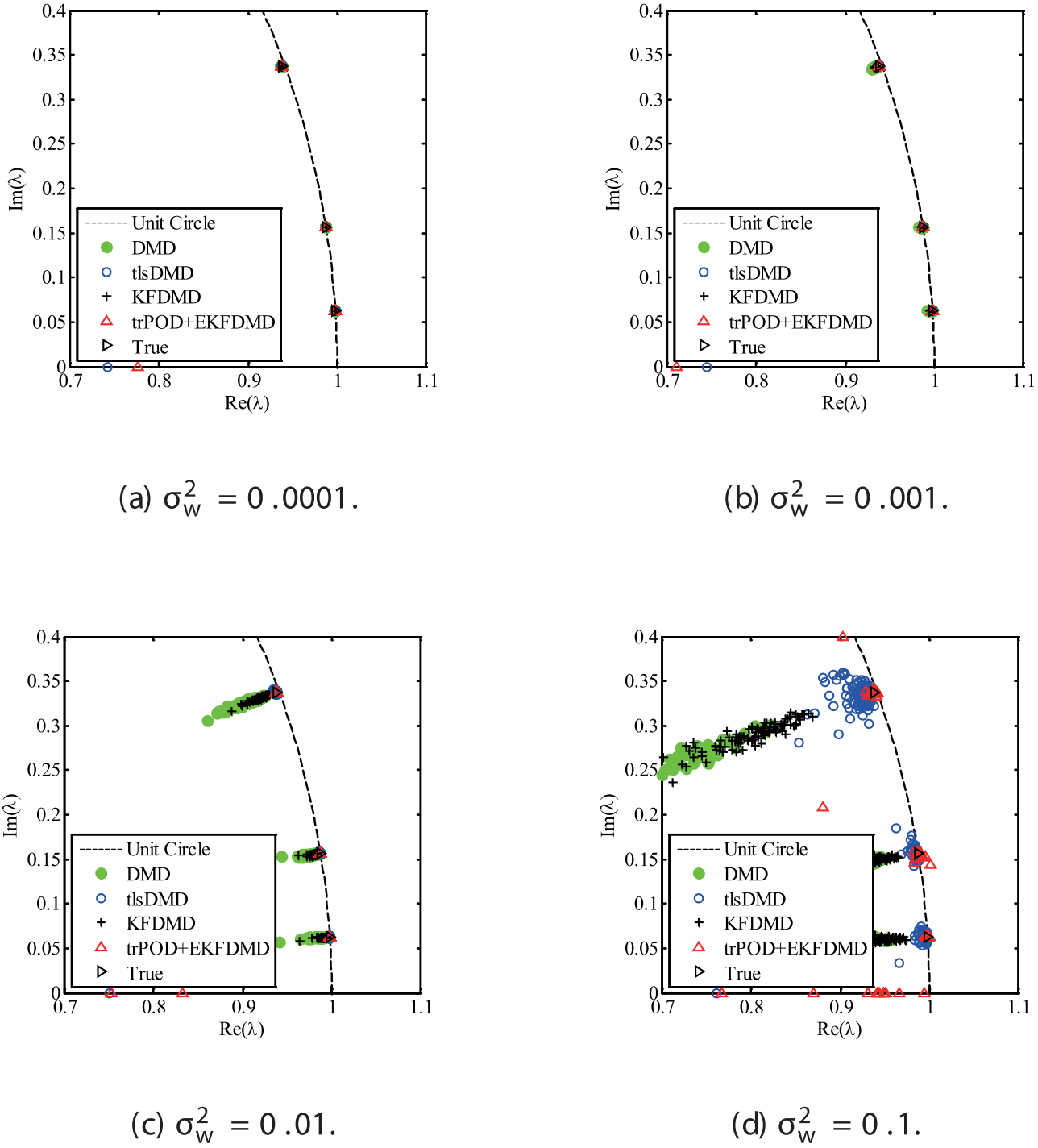
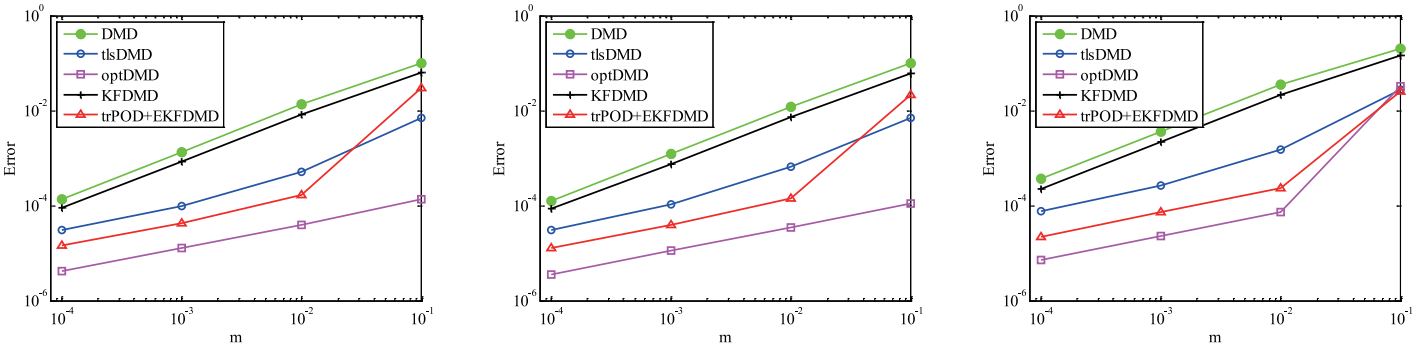


Fig 27. Eigenvalues for multiple runs of a problem with a moderate number of DoFs without system noise, where the seed for the random number is different for multiple runs. Here, rank r is set to be 10.

https://doi.org/10.1371/journal.pone.0209836.g027

performance for trPOD+EKFDMD is not found in this case, together with the results later shown herein. Then, the reconstructed data are shown in Fig 38. Fig 38 illustrates that DMD, KFDMD, and tlsDMD fail to capture the behavior of original data while optDMD works reasonably but sometimes fails to capture the behavior around peaks. Even if we apply the POD



(a) λ_1. (b) λ_2. (c) λ_3.

Fig 28. Errors in the eigenvalues for multiple runs of a problem with a moderate number of DoFs without system noise for the case in which $\sigma_v^2 = 0.1\sigma_w^2$.

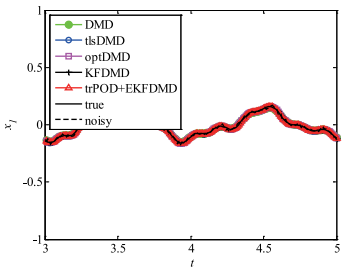
https://doi.org/10.1371/journal.pone.0209836.g028

decomposition, the data reconstructed by trPOD+EKFDMD agree the best with original data. The error in reconstructed data is shown in Fig 39. As shown earlier, the error of EKFDMD is the smallest in the algorithm investigated, similar to the small DoFs problem. Thus, trPOD+-EKFDMD works well to reconstruct the data especially for the case in which system noise is present, even in the moderate number of DoFs problem.

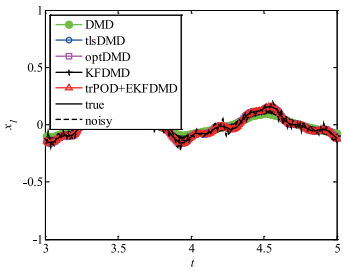
4.4.1 Effects of POD truncation. Similar to the cases without system noise, the effects of the rank number chosen by the user are investigated. Here, $r = 6$ and $r = 20$ are investigated, where the previous standard cases are computed with $r = 10$, as noted earlier. The errors in the eigenvalues estimated with a truncated PODs of $r = 6$ and $r = 20$ and the errors in the reconstructed data with a truncated POD of $r = 6$ and $r = 20$ are shown in Figs 40, 41, 42 and 43. These plots are similar to those with a truncated POD of $r = 10$, which indicates that the rank for the POD truncation does not significantly affect the results for the case in which system noise is present.

4.5 Application to a fluid problem

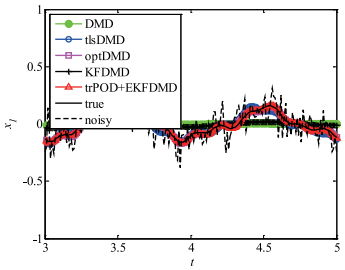
The simulation of a two-dimensional flow around a cylinder is conducted. The Mach number of the freestream velocity is set to be 0.3, and the Reynolds number based on the freestream velocity and the cylinder diameter is set to be 300. For the analysis, LANS3D, [25] which is an in-house compressible fluid solver, is adopted. A cylindrical computational mesh is used, with the numbers of the radial- and azimuthal-direction grid points being 250 and 111, respectively. A compact difference scheme [26] of the sixth order of accuracy is used for spatial derivatives and a second-order backward differencing scheme converged by an alternative-directional-implicit symmetric-Gauss-Seidel method [27, 28] is used for time integration. See Reference [29] for further details. The origin point is set to be the center of the cylinder, and a resolved region (where the mesh density is finer) is set to be inside $10d$ far from the origin point. Here, d is the diameter of the cylinder. For any DMD analyses, the quasi-steady flow data at $x = [0, 10d]$, $y = [-5d, 5d]$, which is in the wake region, are used. The data are mapped to an equally distributed 100×100 mesh. The DMD analyses processed 500 samples of five flow-through data with or without adding observation noise of $\mathcal{N}(0, \sigma_w^2)$, whereas the variance (σ_w^2) is set to be 0.02. In the EFKDMD algorithm, the diagonal parts of the covariance matrix are initially set to be 10^3, similar to previous problems. The diagonal elements of Q and R are set to be 0 and 0.02, respectively, while nondiagonal elements of Q and R are set to be 0.



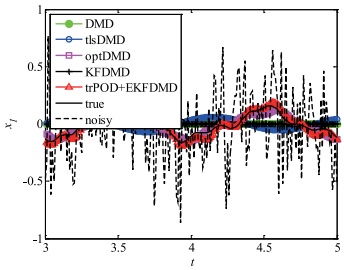
(a) $\sigma_w^2 = 0.0001$.



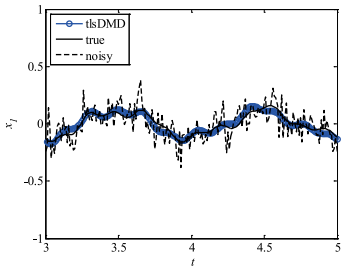
(b) $\sigma_w^2 = 0.001$.



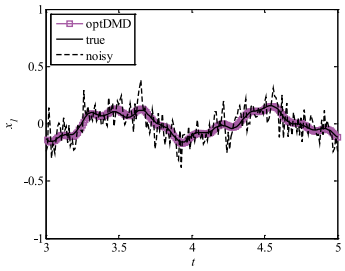
(c) $\sigma_w^2 = 0.01$.



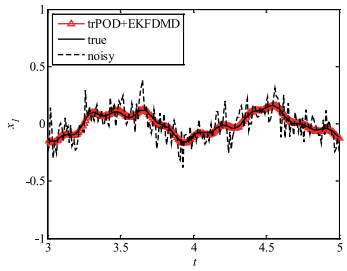
(d) $\sigma_w^2 = 0.1$.



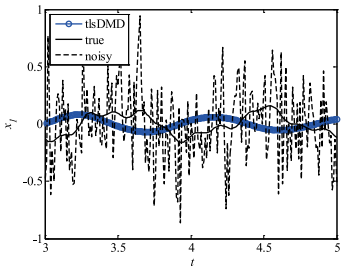
(e) $\sigma_w^2 = 0.01$, tlsDMD.



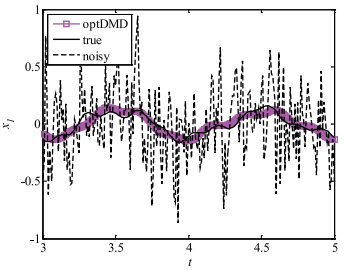
(f) $\sigma_w^2 = 0.01$, optDMD.



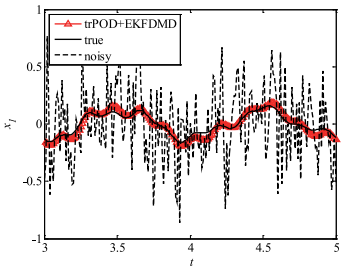
(g) $\sigma_w^2 = 0.01$, trPOD+EKFDMD.



(h) $\sigma_w^2 = 0.1$, tlsDMD.



(i) $\sigma_w^2 = 0.1$, optDMD.



(j) $\sigma_w^2 = 0.1$, trPOD+EKFDMD.

Fig 29. Reconstructed data of the first node for a problem with a moderate number of DoFs without system noise.

https://doi.org/10.1371/journal.pone.0209836.g029

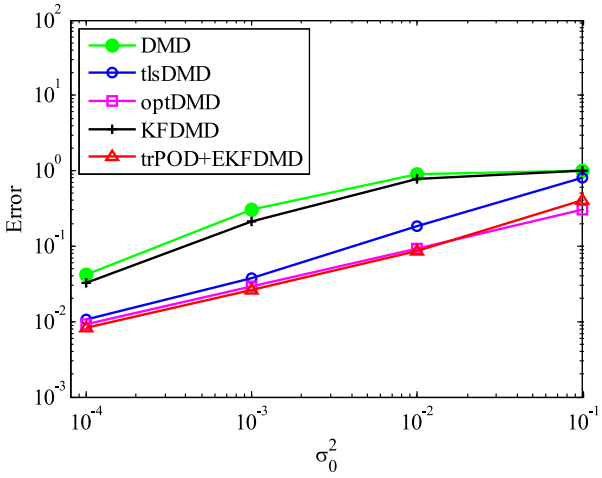
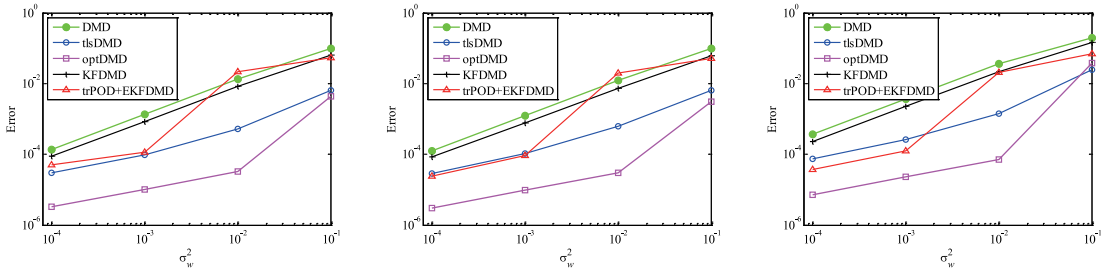


Fig 30. Errors in the reconstructed data for multiple runs of a problem with a moderate number of DoFs without system noise. Here, rank *r* is set to be 10.

https://doi.org/10.1371/journal.pone.0209836.g030



(a) λ_1. (b) λ_2. (c) λ_3.

Fig 31. Errors in the eigenvalues for multiple runs of a problem with a moderate number of DoFs without system noise, whereas the rank *r* is set to be 6. Here the seed for the random numbers is different for multiple runs.

https://doi.org/10.1371/journal.pone.0209836.g031

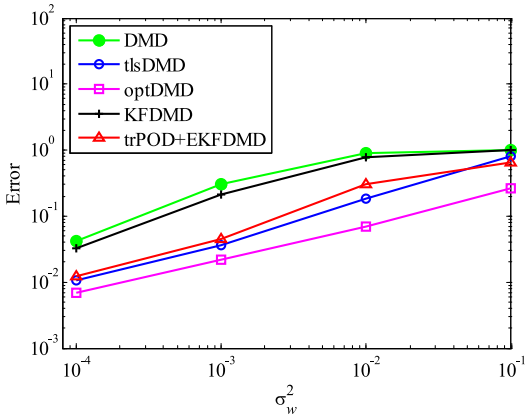
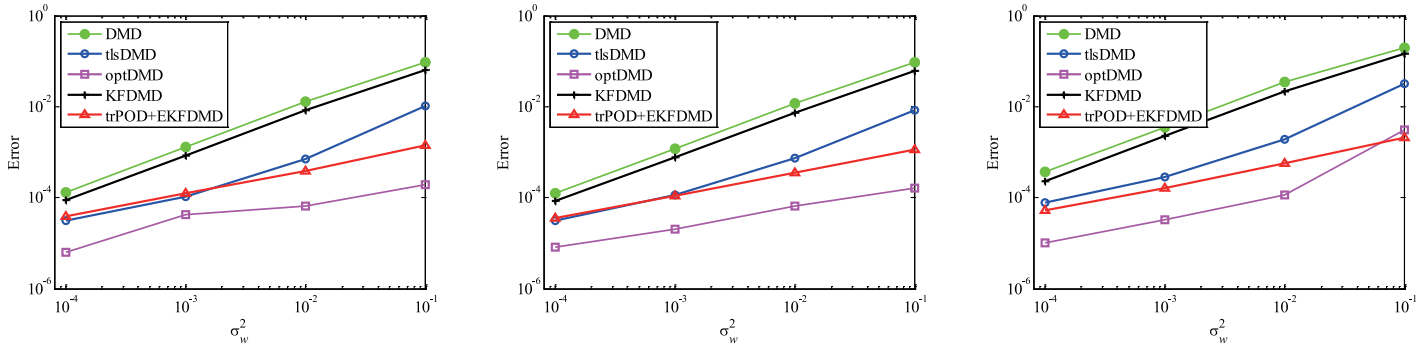


Fig 32. Errors in the reconstructed data for multiple runs of a problem with a moderate number of DoFs without system noise, whereas the rank *r* is set to be 6. Here, the seed for the random numbers is different for multiple runs.

https://doi.org/10.1371/journal.pone.0209836.g032



(a) λ_1. (b) λ_2. (c) λ_3.

Fig 33. Errors in the eigenvalues for multiple runs of a problem with a moderate number of DoFs without system noise, whereas the rank *r* is set to be 20. Here, the seed for the random number is different for multiple runs.

https://doi.org/10.1371/journal.pone.0209836.g033

First, the results without noise are processed by DMD, tlsDMD and KFDMD, where KFDMD adopts the truncated POD (Eq 46) as a preconditioner. The eigenvalues computed by the DMD, tlsDMD and trPOD+KFDMD methods are shown in Fig 44. The eigenvalues computed by KFDMD agree well with those of the standard DMD. The lowest frequencies computed by DMD and KFDMD correspond to the Strouhal number $St = fd/u_\infty \sim 0.2$, which is a well-known characteristic frequency for the Kármán vortex street of a cylinder wake, where f and u_∞ are the frequency and the freestream velocity, respectively.

Then, the data with noise are processed. The snapshot data of the instantaneous flow field are shown in Fig 45. Flow fields filtered using only trPOD are shown in Fig 46. The noise can be reduced using trPOD. These 30-DoF data are used for KFDMD analyses.

Fig 47, which illustrates the eigenvalues of DMD, tlsDMD, and EKFDMD, shows that the EKFDMD results are better than the results of the standard DMD and tlsDMD. Here,

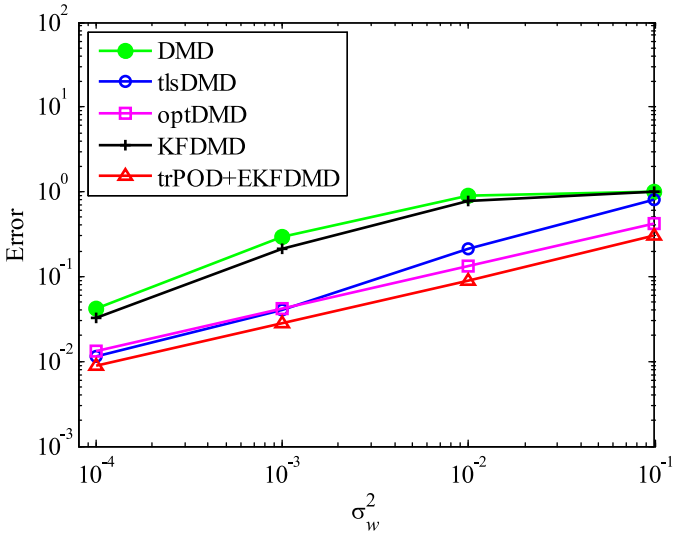
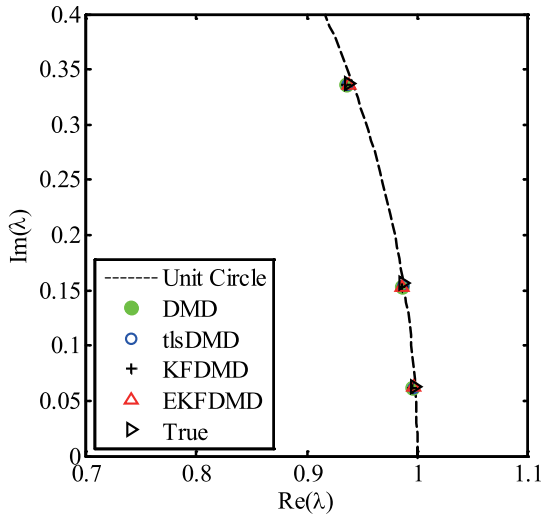
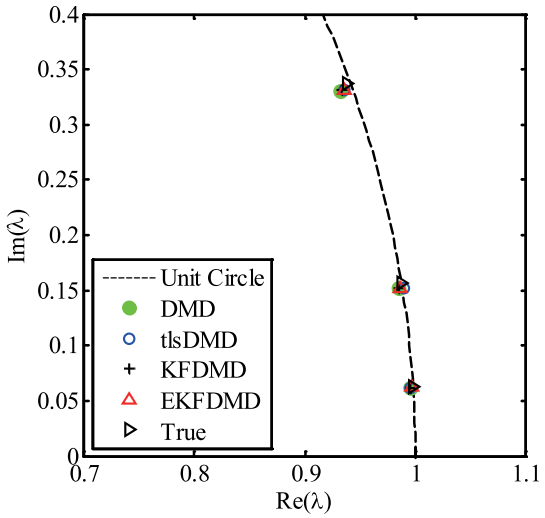


Fig 34. Errors in the reconstructed data for multiple runs of a problem with a moderate number of DoFs without system noise, whereas the rank *r* is set to be 20. Here, the seed for the random number is different for multiple runs.

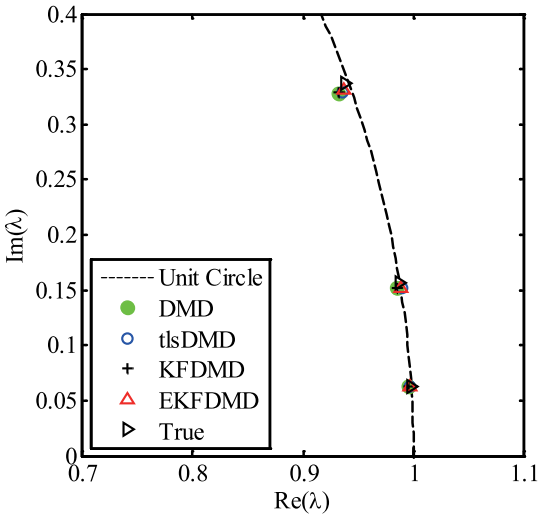
https://doi.org/10.1371/journal.pone.0209836.g034



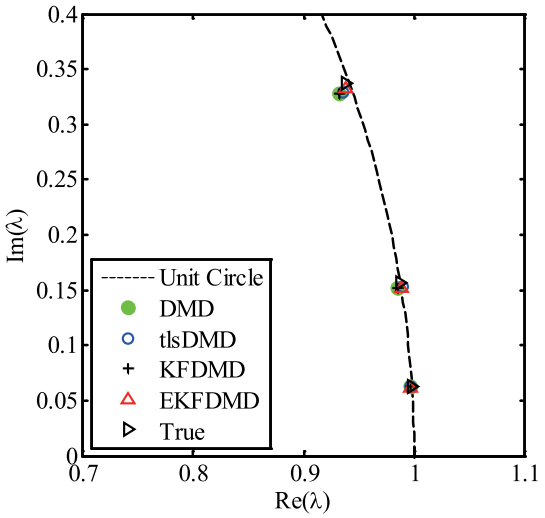
(a) $\sigma_w^2 = 0.0001$.



(b) $\sigma_w^2 = 0.001$.



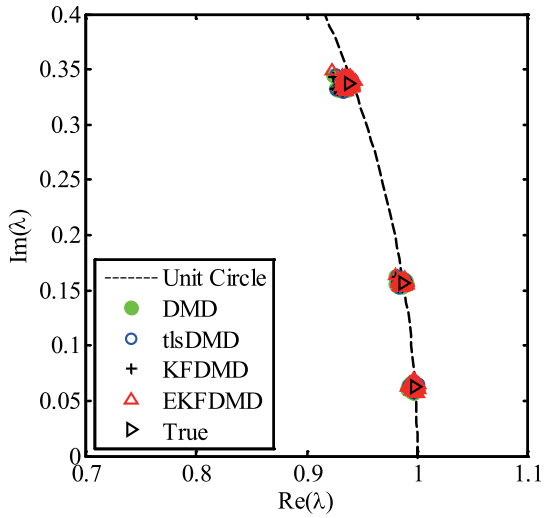
(c) $\sigma_w^2 = 0.01$.



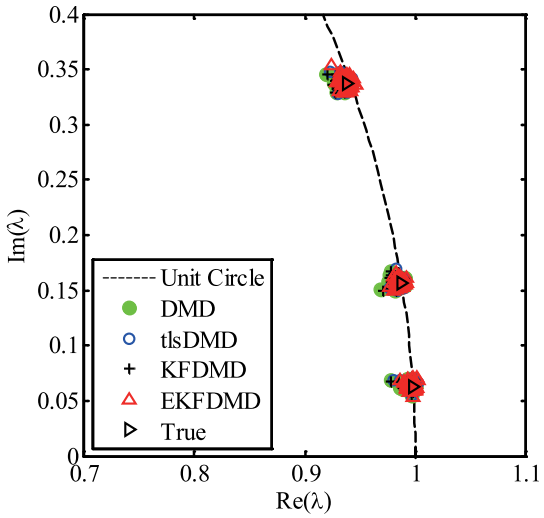
(d) $\sigma_w^2 = 0.1$.

Fig 35. Eigenvalues for a problem with a moderate number of DoFs with system noise. The results of all algorithms are almost identical in this plot. Here, rank r is set to be 10.

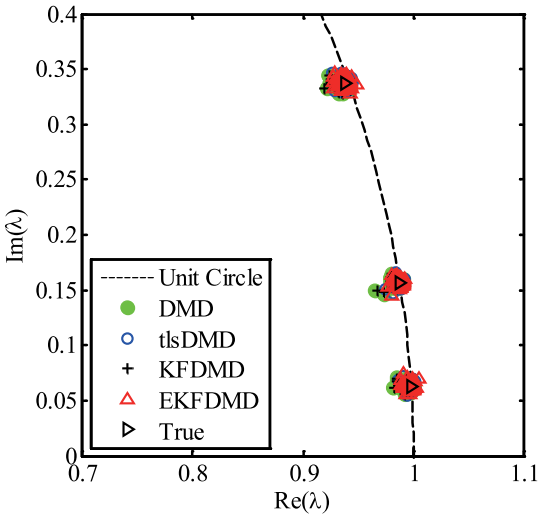
https://doi.org/10.1371/journal.pone.0209836.g035



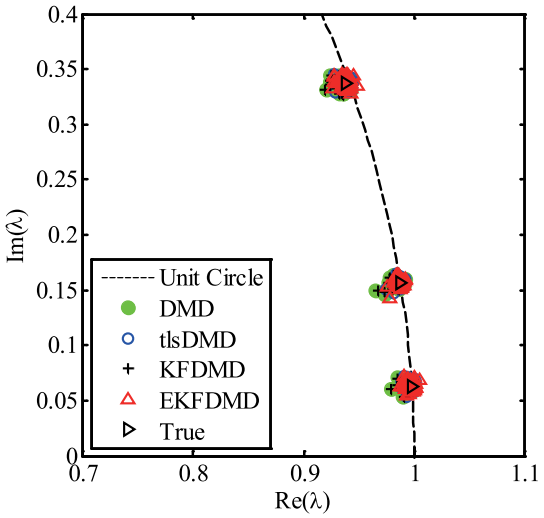
(a) $\sigma_w^2 = 0.0001$.



(b) $\sigma_w^2 = 0.001$.



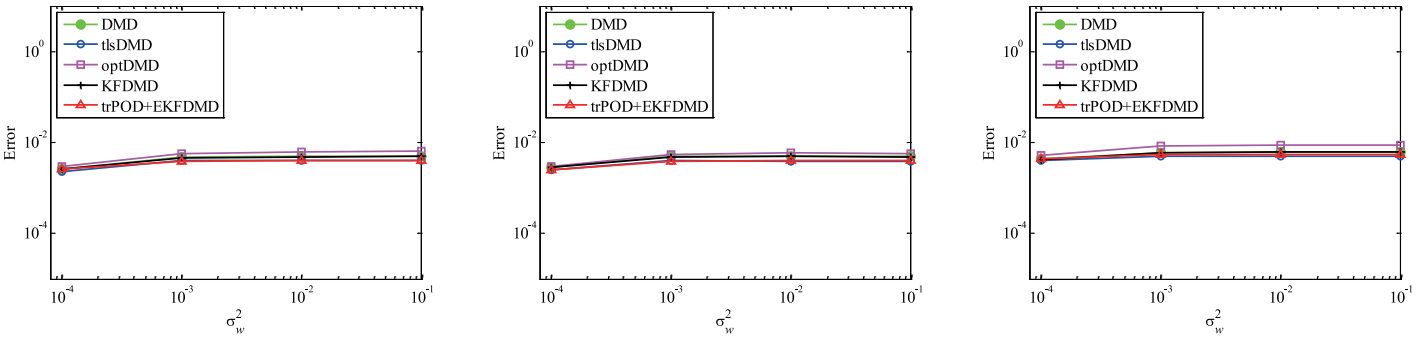
(c) $\sigma_w^2 = 0.01$.



(d) $\sigma_w^2 = 0.1$.

Fig 36. Eigenvalues for multiple runs of a problem with a moderate number of DoFs with system noise. The results of all algorithms are almost identical in this plot. Here, rank r is set to be 10.

https://doi.org/10.1371/journal.pone.0209836.g036



(a) λ_1. (b) λ_2. (c) λ_3.

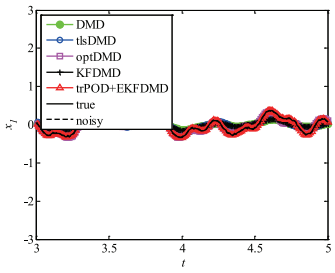
Fig 37. Errors in the in the eigenvalues for multiple runs of a problem with a moderate number of DoFs with system noise, where the seed for random numbers is different for multiple runs. Here, rank r is set to be 10.

https://doi.org/10.1371/journal.pone.0209836.g037

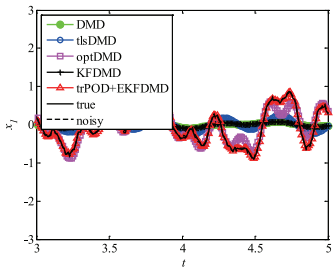
trPOD+EKFDMD accurately predicts from the steady flow mode (eigenvalue of unity) up to the fifth oscillation mode, which corresponds to six points on the upper half of the unit circle. In addition, it should be noted that the strength of EKFDMD is that the data are denoised once. Fig 48 shows the mode histories of trPOD modes 2, 4, 6, and 8. The histories of modes 2 and 4 are approximately the same for noisy data and EKFDMD combined with the trPOD preconditioner, because these modes are strong enough compared with the noise level. On the other hand, the histories of modes 6 and 8 are cleaned up well. Finally, the flow fields of denoised data (in this case, the temporal coefficients of the trPOD modes are filtered) are shown in Fig 49, and the data are slightly further cleaned up compared to the results obtained only with trPOD, as shown in Fig 46.

5 Complexity and computational cost

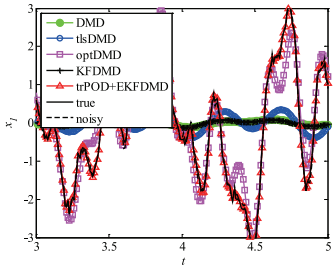
In this section, the complexity and computational cost of EKFDMD are discussed. Here, multiplication for single elements is assumed to have a complexity of $O(1)$, and the multiplication of matrices of size of $l \times m$ and $m \times n$ is estimated to be $O(lmn)$ under the dense matrix computation. In the EKFDMD procedure, except when using trPOD as a preconditioner, the main computational cost comes from Eqs 36 and 37 for the prediction step and from Eqs 39, 38, 40 and 43 for the updating step. For each step, the computational complexity is summarized in Table 1. In total, the most significant complexity is considered to be $O(n^6)$ for one step. Therefore, if we have m samples, then the computational complexity for m-time steps becomes $O(mn^6)$. The complexity and the required memory of EKFDMD are compared with those of the other algorithms in Table 2, where estimation of the complexities of DMD and online DMD in the previous study [14] are adopted, and the complexity of KFDMD is estimated in the present study. In addition, Fig 50 shows the computational time for 500 samples with different DoF problems. The Matlab software is used with Intel Xeon E5620 2.4GHz processor. The computational time is averaged over 20 runs for the small size of $m < 50$, while it is not for the large size but the repeatability is confirmed. Both Table 2 and Fig 50 show that EKFDMD requires significant computational cost, and applying trPOD as a preconditioner is strongly recommended for the practical use of EKFDMD. In practical use, matrices F and H for EKFDMD are sparse and the corresponding computational cost and memory of EKFDMD can be decreased by using implementations of routines for the sparse matrix in the software



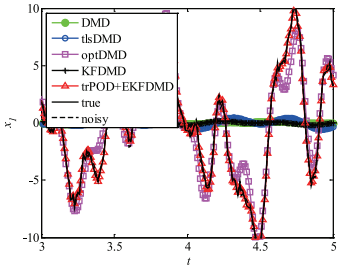
(a) $\sigma_w^2 = 0.0001$.



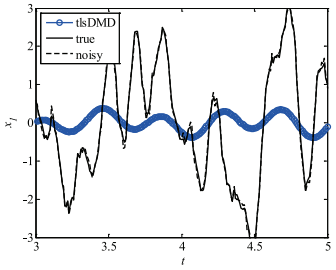
(b) $\sigma_w^2 = 0.001$.



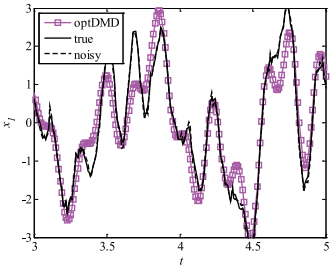
(c) $\sigma_w^2 = 0.01$.



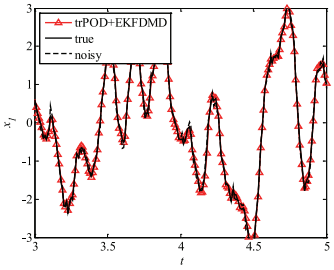
(d) $\sigma_w^2 = 0.1$.



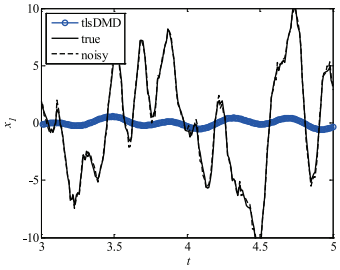
(e) $\sigma_w^2 = 0.01$, tlsDMD.



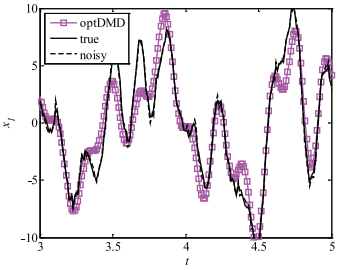
(f) $\sigma_w^2 = 0.01$, optDMD.



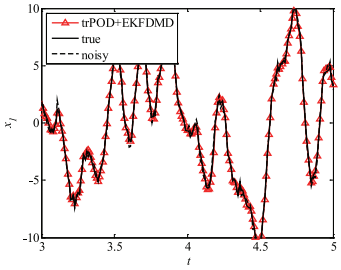
(g) $\sigma_w^2 = 0.01$, trPOD+EKFDMD.



(h) $\sigma_w^2 = 0.1$, tlsDMD.



(i) $\sigma_w^2 = 0.1$, optDMD.



(j) $\sigma_w^2 = 0.1$, trPOD+EKFDMD.

Fig 38. Reconstructed data of the first node for a problem with a moderate number of DoFs with system noise. Here, rank r is set to be 10.

https://doi.org/10.1371/journal.pone.0209836.g038

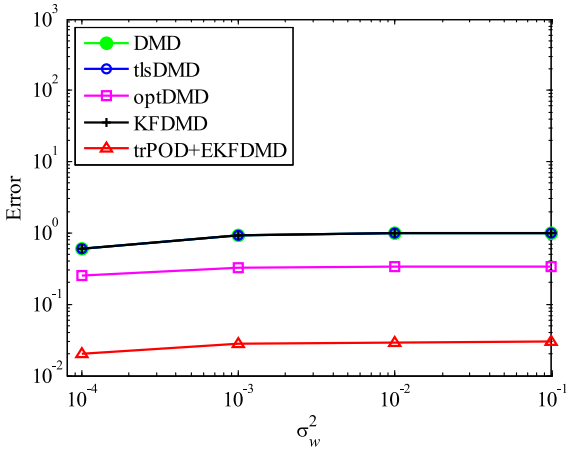
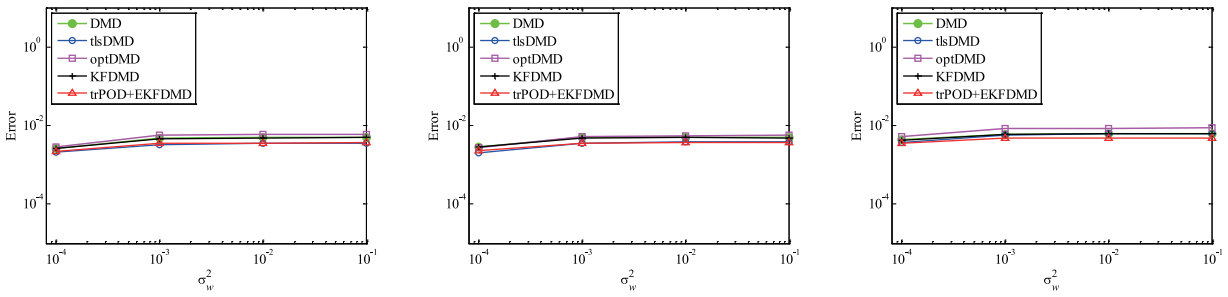


Fig 39. Errors in the reconstructed data for multiple runs of a problem with a moderate number of DoFs with system noise. Here, rank r is set to be 10.

https://doi.org/10.1371/journal.pone.0209836.g039



(a) λ_1. (b) λ_2. (c) λ_3.

Fig 40. Errors in the eigenvalues for multiple runs of a problem with a moderate number of DoFs with system noise for the case in which rank r is set to be 6. The algorithms are almost identical.

https://doi.org/10.1371/journal.pone.0209836.g040

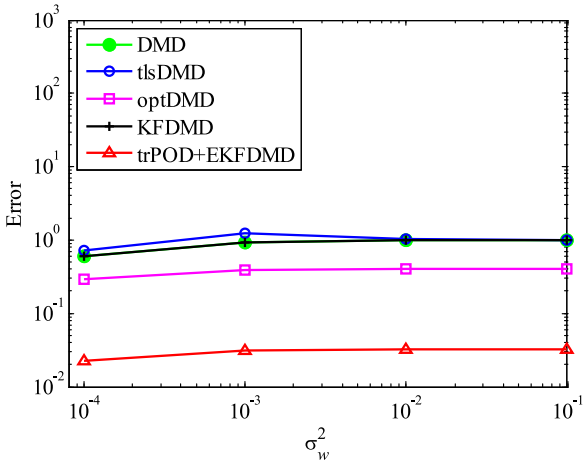
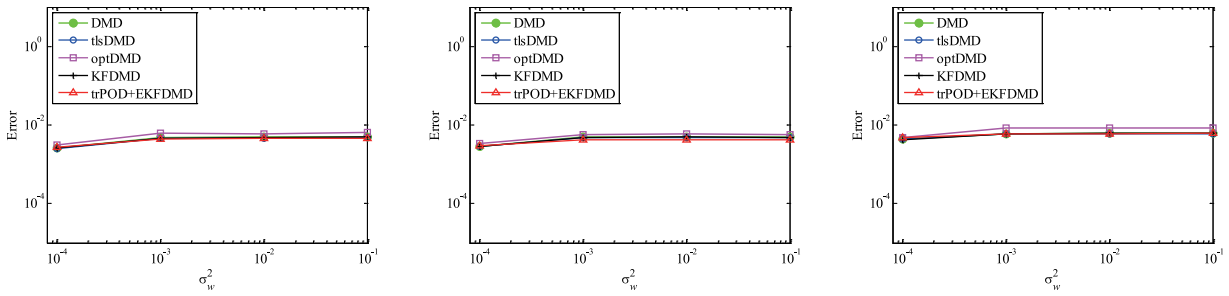


Fig 41. Errors in the reconstructed data for multiple runs of a problem with a moderate number of DoFs without system noise for the case in which rank r is set to be 6, where the seed for random numbers is different for multiple runs.

https://doi.org/10.1371/journal.pone.0209836.g041



(a) λ_1. (b) λ_2. (c) λ_3.

Fig 42. Errors in the eigenvalues for multiple runs of a problem with a moderate number of DoFs with system noise for the case in which rank *r* is set to be 20, where the seed for random numbers is different for multiple runs.

https://doi.org/10.1371/journal.pone.0209836.g042

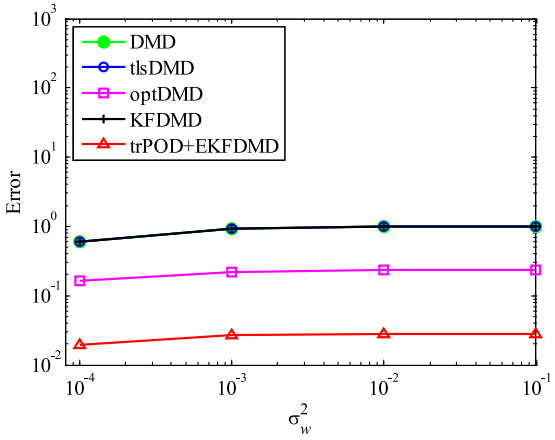


Fig 43. Errors in the reconstructed data for multiple runs of a problem with a moderate number of DoFs with system noise for the case in which rank *r* is set to be 20, where the seed for random numbers is different for multiple runs.

https://doi.org/10.1371/journal.pone.0209836.g043

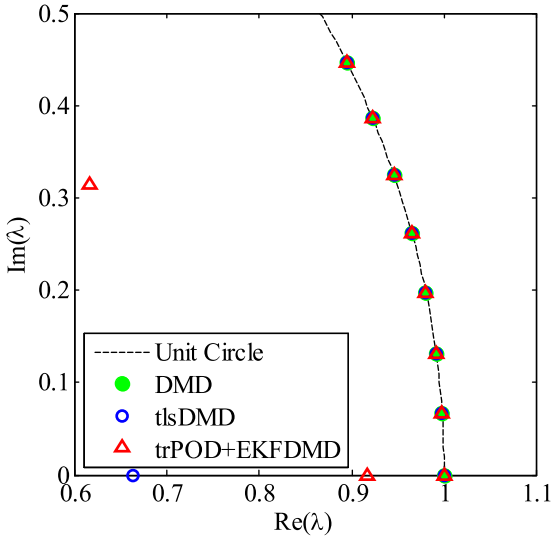


Fig 44. Eigenvalues for a flow problem without noise.

https://doi.org/10.1371/journal.pone.0209836.g044

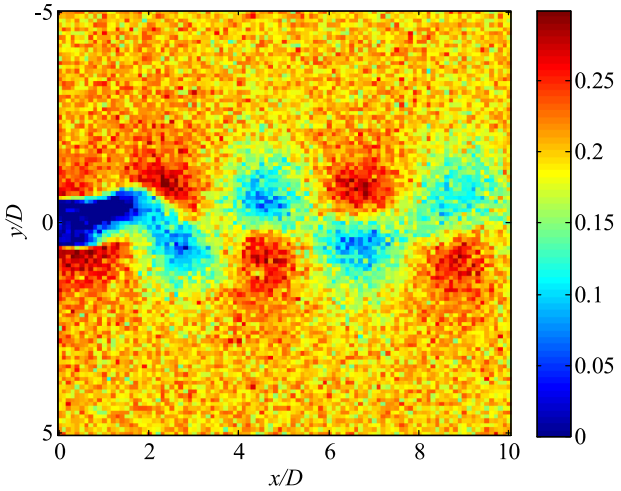


Fig 45. Noisy flow field data processed by several DMD methods. The *x*-direction velocity is visualized, where the freestream velocity is set to be 0.3.

https://doi.org/10.1371/journal.pone.0209836.g045

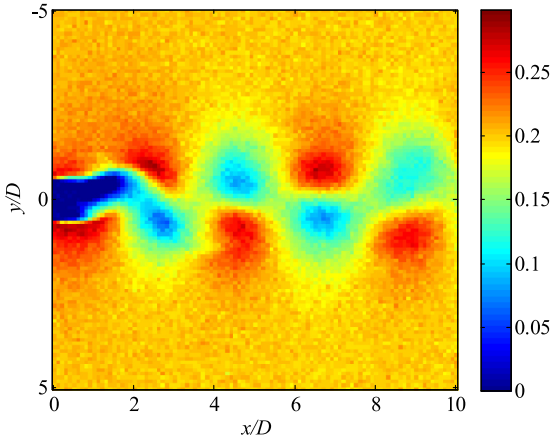


Fig 46. trPOD 30-mode reconstruction of flow fields. The *x*-direction velocity is visualized, where the freestream velocity is set to be 0.3.

https://doi.org/10.1371/journal.pone.0209836.g046

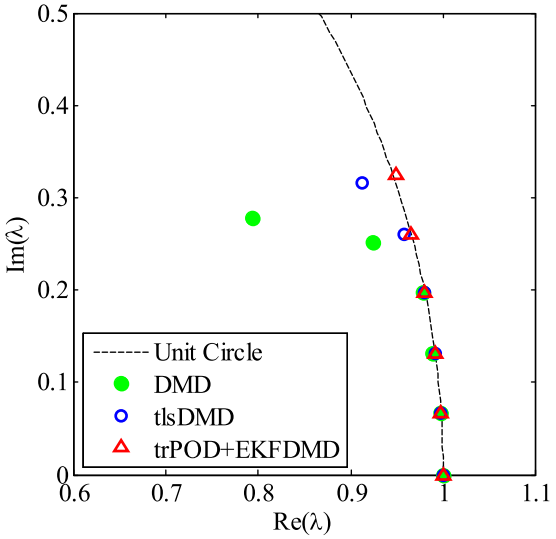


Fig 47. Eigenvalues for a flow problem with noise.

https://doi.org/10.1371/journal.pone.0209836.g047

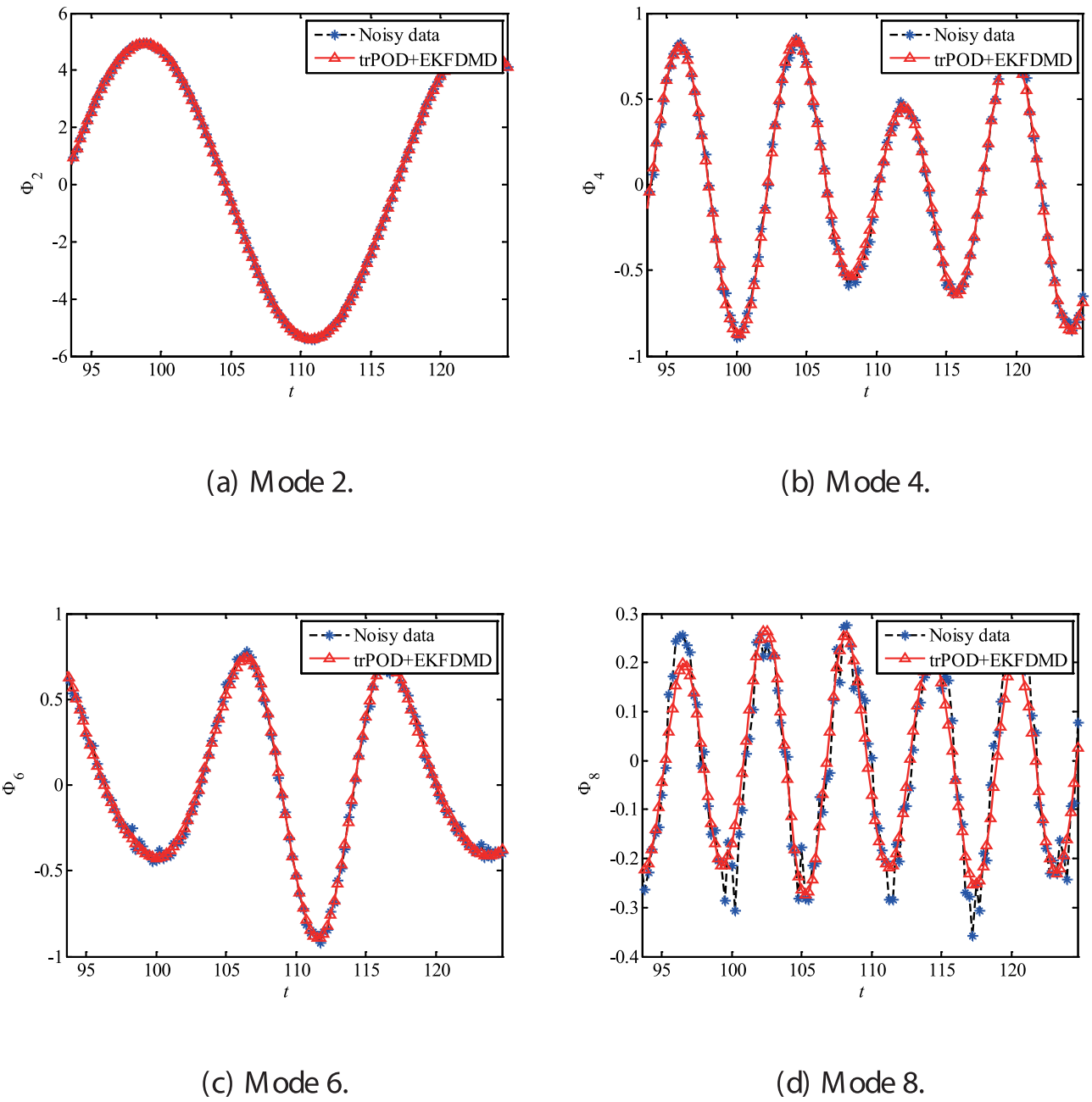


Fig 48. Time histories of POD modes 2, 4, 6, and 8 of the data of the flow problem.

https://doi.org/10.1371/journal.pone.0209836.g048

utilized as we did. However, the complexity of EKFDMD is still higher with the routines for the sparse matrix than the other algorithms as shown in Fig 50.

Conclusions

A dynamic mode decomposition method based on the extended Kalman filter (EKFDMD) was proposed for simultaneous parameter estimation and denoising. The numerical experiments of the present study reveal that the proposed method can estimate the eigenstructure of

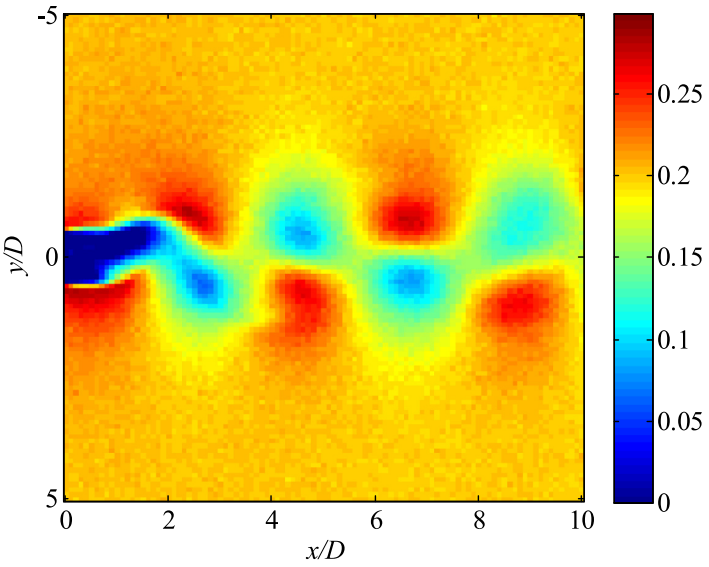


Fig 49. EKFDMD-filtered flow fields. The x-direction velocity is visualized, where the freestream velocity is set to be 0.3.

https://doi.org/10.1371/journal.pone.0209836.g049

Table 1. Computational time for each procedure in KFDMD.

procedure	equation	complexity
predicting step	Eq 36	$O(n^2)$
	Eq 37	$O((n+n^2)^3)$
updating step	Eq 39	$2O((n+n^2)^2\, n)$
	Eq 38	$O(n^3) + O(n^2(n+n^2)) + O((n+n^2)^2\, n)$
	Eq 40	$2O(n(n+n^2))$
	Eq 43	$2O(n(n+n^2)^2) + O((n+n^2)^3)$

https://doi.org/10.1371/journal.pone.0209836.t001

the matrix A better than or as well as existing algorithms in its online procedure for a problem with a small number of DoFs, whereas EKFDMD simultaneously denoises the data. In particular, the EKFDMD works better for data reconstruction in the case in which the system noise is present than existing algorithms. However, this algorithm has the drawback of computational cost. This drawback is addressed by preconditioning of truncated POD (trPOD), thouth it prevents the algorithm from being fully online. Then, EKFDMD with trPOD is applied to a problem with a moderate number of DoFs and a fluid system. The performance of EKFDMD is slightly degraded by decreasing the rank number of trPOD in the case without system noise

Table 2. Comparison of complexity and memory for m-sample computation for the estimation in the final time step once.

algorithm	computational time	memory
DMD	$O(mn^2)$	mn
online DMD	$O(mn^2)$	$2n^2$
KFDMD without trPOD (fast algorithm)	$O(mn^2)$	$2n^2$
EKFDMD without trPOD	$O(mn^6)$	$(n+n^2)(n+n^2+1)$

https://doi.org/10.1371/journal.pone.0209836.t002

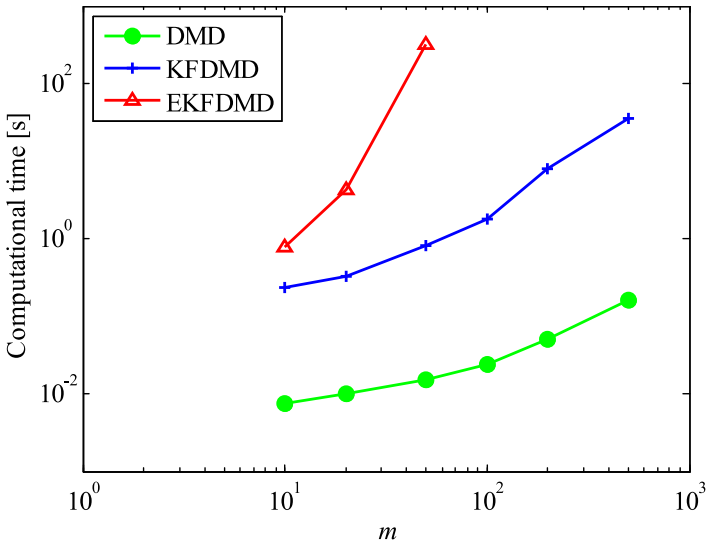


Fig 50. Computational time for DMD, KFDMD, and EKFDMD.

https://doi.org/10.1371/journal.pone.0209836.g050

while the performance does not change in the case with system noise with regardless of the rank number. It should be noted that all the performance of EKFDMD is preferable in the analysis of noisy data.

Supporting information

S1 Video. Video for original noisy data of cylinder flow. The video corresponding to Fig 45.
(MP4)

S2 Video. Video for trPOD(30) filtered data of cylinder flow. The video corresponding to Fig 46.
(MP4)

S3 Video. Video for EKFDMD filtered data of cylinder flow. The video corresponding to Fig 49.
(MP4)

Author Contributions

Conceptualization: Taku Nonomura, Hisaichi Shibata.

Data curation: Taku Nonomura, Hisaichi Shibata.

Formal analysis: Taku Nonomura.

Funding acquisition: Ryoji Takaki.

Supervision: Ryoji Takaki.

Writing – original draft: Taku Nonomura.

Writing – review & editing: Taku Nonomura, Hisaichi Shibata, Ryoji Takaki.

References

1. Taira K, Brunton SL, Dawson ST, Rowley CW, Colonius T, McKeon BJ, et al. Modal analysis of fluid flows: An overview. AIAA Journal. 2017; 55(12):4013–4041. https://doi.org/10.2514/1.J056060
2. Rowley CW, Colonius T, Murray RM. Model reduction for compressible flows using POD and Galerkin projection. Physica D: Nonlinear Phenomena. 2004; 189(1-2):115–129. https://doi.org/10.1016/j.physd.2003.03.001
3. Berkooz G, Holmes P, Lumley LJ. The proper orthogonal decomposition in the analysis of turbulent flows. Annual Review of Fluid Mechanics. 1993; 25(1971):539–575. https://doi.org/10.1146/annurev.fl.25.010193.002543
4. Theofilis V. Global Linear Instability. Annual Review of Fluid Mechanics. 2011; 43(1):319–352. https://doi.org/10.1146/annurev-fluid-122109-160705
5. Shibata H, Ohmichi Y, Watanabe Y, Suzuki K. Global stability analysis method to numerically predict precursor of breakdown voltage. Plasma Sources Science and Technology. 2015; 24(5). https://doi.org/10.1088/0963-0252/24/5/055014
6. Ohmichi Y, Suzuki K. Assessment of global linear stability analysis using a time-stepping approach for compressible flows. International Journal for Numerical Methods in Fluids. 2016; 80(10):614–627. https://doi.org/10.1002/fld.4166
7. Schmid PJ. Dynamic mode decomposition of numerical and experimental data. Journal of Fluid Mechanics. 2010; 656(July 2010):5–28.
8. Tu JH, Rowley CW, Luchtenburg DM, Brunton SL, Kutz JN. On dynamic mode decomposition: Theory and applications. Journal of Computational Dynamics. 2014; 1(2):391–421. https://doi.org/10.3934/jcd.2014.1.391
9. Wang L, Feng LH. Extraction and Reconstruction of Individual Vortex-Shedding Mode from Bistable Flow. AIAA Journal. 2017; p. 1–13.
10. Priebe S, Tu JH, Rowley CW, Martín MP. Low-frequency dynamics in a shock-induced separated flow. Journal of Fluid Mechanics. 2016; 807:441–477.
11. Ohmichi Y. Preconditioned dynamic mode decomposition and mode selection algorithms for large datasets using incremental proper orthogonal decomposition. AIP Advances. 2017; 7(7):075318. https://doi.org/10.1063/1.4996024
12. Dawson ST, Hemati MS, Williams MO, Rowley CW. Characterizing and correcting for the effect of sensor noise in the dynamic mode decomposition. Experiments in Fluids. 2016; 57(3):42. https://doi.org/10.1007/s00348-016-2127-7
13. Hemati MS, Rowley CW, Deem EA, Cattafesta LN. De-biasing the dynamic mode decomposition for applied Koopman spectral analysis of noisy datasets. Theoretical and Computational Fluid Dynamics. 2017; p. 1–20.
14. Zhang H, Rowley CW, Deem EA, Cattafesta LN. Online dynamic mode decomposition for time-varying systems. arXiv preprint arXiv:170702876. 2017.
15. Nonomura T, Shibata H, Takaki R. Dynamic mode decomposition using a Kalman filter for parameter estimation. AIP Advances 8, 2018; 8(105106).
16. Kalman RE. A New Approach to Linear Filtering and Prediction Problems. Journal of Basic Engineering. 1960; 82(1):35. https://doi.org/10.1115/1.3662552
17. Surana A, Banaszuk A. Linear observer synthesis for nonlinear systems using Koopman operator framework. IFAC-PapersOnLine. 2016; 49(18):716–723. https://doi.org/10.1016/j.ifacol.2016.10.250
18. Surana A, Williams MO, Morari M, Banaszuk A. Koopman operator framework for constrained state estimation. In: Decision and Control (CDC), 2017 IEEE 56th Annual Conference on. IEEE; 2017. p. 94–101.
19. Kutz JN, Brunton SL, Brunton BW, Proctor JL. Dynamic mode decomposition: data-driven modeling of complex systems. vol. 149. SIAM; 2016.
20. Askham T, Kutz JN. Variable projection methods for an optimized dynamic mode decomposition. SIAM Journal on Applied Dynamical Systems. 2018; 17(1):380–416. https://doi.org/10.1137/M1124176
21. Hemati M, Deem E, Williams M, Rowley CW, Cattafesta LN. Improving separation control with noise-robust variants of dynamic mode decomposition. In: AIAA-Paper 2016-1103; 2016. p. 1103.
22. Jovanović MR, Schmid PJ, Nichols JW. Sparsity-promoting dynamic mode decomposition. Physics of Fluids. 2014; 26(2):1–22.
23. Rowley CW. A library of tools for computing variants of Dynamic Mode Decomposition; 2017. https://github.com/cwrowley/dmdtools/tree/master/.
24. Askham T. duqbo/optdmd: optdmd v1.0.0; 2017. https://doi.org/10.5281/zenodo.439385.

25. Fujii K, Endo H, Yasuhara M. Activities of Computational Fluid Dynamics in Japan: Compressible Flow Simulations. High Performance Computing Research and Practice in Japan, Wiley Professional Computing, JOHN WILEY& SONS. 1990; p. 139–161.
26. Lele SK. Compact Finite Difference Schemes with Spectral-like Resolution. Journal of Computational Physics. 1992; 103(1):16–42. https://doi.org/10.1016/0021-9991(92)90324-R
27. Fujii K. Efficiency Improvement of Unified Implicit Relaxation/Time Integration Algorithms. AIAA Journal. 1999; 37(1):125–128.
28. Nishida H, Nonomura T. ADI-SGS Scheme on Ideal Magnetohydrodynamics. Journal of Computational Physics. 2009; 228:3182–3188. https://doi.org/10.1016/j.jcp.2009.01.032
29. Sato M, Nonomura T, Okada K, Asada K, Aono H, Yakeno A, et al. Mechanisms for laminar separated-flow control using dielectric-barrier-discharge plasma actuator at low Reynolds number. Physics of Fluids. 2015; 27:1–29. https://doi.org/10.1063/1.4935357