Data-driven Distributed Learning of Multi-agent Systems: A Koopman Operator Approach

Sai Pushpak Nandanoori, Seemita Pal, Subhrajit Sinha, Soumya Kundu, Khushbu Agarwal, Sutanay Choudhury
Pacific Northwest National Laboratory, Richland, WA 99354
Emails: {saipushpak.n, seemita.pal, subhrajit.sinha, soumya.kundu, khushbu.agarwal, sutanay.choudhury}@pnnl.gov

***Abstract*—Koopman operator theory provides a model-free technique for studying nonlinear dynamical systems purely from data. Since the Koopman operator is infinite-dimensional, researchers have developed several methods that provide a finite-dimensional approximation of the Koopman operator so that it can be applied for practical use cases. One common thing with most of the methods is that their solutions are obtained by solving a centralized minimization problem. In this work, we treat the dynamical system to be a multi-agent system and propose an algorithm to compute the finite-dimensional approximation of the Koopman operator in a distributed manner using the knowledge of the topology of the underlying multi-agent system. The proposed distributed approach is shown to be equivalent to the centralized learning problem and results in a sparse Koopman whose block structure mimics the Laplacian of the multi-agent system. Extensive simulation studies illustrate the proposed framework on the network of oscillators and the IEEE 68 bus system.**

I. Introduction

Traditional analysis of nonlinear dynamical systems that exhibit rich and complex behavior such as multiple equilibrium points, periodic orbits, chaotic attractors, limit cycles, etc., assumes the knowledge of the governing equations [1]. In contrast to the study of linear dynamical systems, the approach to study each nonlinear system can be unique and hence challenging, especially for higher-order nonlinear dynamical systems. Koopman operator theory (KOT) provides an alternate approach to study nonlinear systems using spatio-temporal time-series data. Instead of studying the evolution of the nonlinear system on the state space, the state space data is transformed into a linear observable space where the evolution of the observables is given by the linear Koopman operator [2]–[4]. However, there is a trade-off because though the lifted dynamical system is linear, it is typically infinite-dimensional. [2]–[5].

The Koopman operator being infinite dimensional is computationally intractable. Recent works proposed numerous methods to identify approximation of the infinite-dimensional Koopman operator on a finite-dimensional space. Most notable works include dynamic mode decomposition (DMD) [4], [6], [7], extended DMD (EDMD) [8], [9], Hankel DMD [10], naturally structured DMD (NS-DMD) [11] and deep learning based DMD (deepDMD) [12]–[14]. These methods are data-driven and one or more of these methods have been successfully applied for system identification [4], [15] including system identification from noisy data [16], data-driven observability/controllability gramians for nonlinear systems [17], [18], control design [19]–[21], data-driven causal inference in dynamical systems [22], [23], data-driven quantification of information transfer [24]–[26] and to identify persistence of excitation conditions for nonlinear systems [27].

The above-mentioned methods are network topology independent and achieve the centralized computation of the finite-dimensional Koopman operator. Three of the existing works that are most relevant to our proposed approach are discussed in more detail. The Sparse DMD introduced in [28] promotes sparsity of the Koopman operator by imposing a ℓ_1 regularization on the Koopman learning problem. Although this method succeeds in capturing the underlying system dynamics, it suffers from two drawbacks: (a) the computational complexity of obtaining the sparse Koopman increases exponentially with an increase in the number of system states, (b) the choice of the weight for the sparsity promoting cost is a user-defined decision variable and this choice determines the extent of the sparsity of the Koopman operator. The compositional Koopman operator introduced in [29] leverages the graph neural networks and imposes additional structural constraints on the Koopman operator on the observable space to obtain a sparse Koopman. Another drawback with the existing works includes retraining the Koopman when the system undergoes any topological changes.

The Koopman operator method in this paper is developed to perform state predictions. Therefore, its predictive performance is evaluated by applying to two different dynamical systems.

Problem statement: Develop a distributed approach for computing the Koopman operator for large-scale networks, while maintaining the accuracy.

Summary of contributions: The main contributions of this work are summarized as follows.

- Develop methods for learning distributed Koopman operator for large-scale networks, using system topology and network sparsity properties.
- Our method scales better than state-of-the-art approaches while maintaining model predicted accuracy when applied for use-cases.

Pacific Northwest National Laboratory (PNNL) is a multi-program national laboratory operated by Battelle for the U.S. Department of Energy (DOE) under contract No. DE-AC05-76RL01830. This work was supported by the DOE's Advanced Grid Modeling (AGM) program.

The paper is organized as follows. The mathematical background for this work is given in Section II and the main results of this work are discussed in Section III. The proposed distributive computation is illustrated on a large-scale network of oscillators and IEEE 68 bus system in Section IV. Finally, concluding remarks are provided in Section V.

II. Mathematical Preliminaries

Consider a multi-agent system evolving over a network/graph ($\mathcal{G}$) defined by the Laplacian L. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where $\mathcal{V} = \{v_1, v_2, \ldots, v_s\}$ and $\mathcal{E} = \{e_1, e_2, \ldots, e_m\}$ denote the set of nodes and edges respectively. We know,

$$L = D - A = EE^\top$$

where $D \in \mathbb{R}^{s\times s}$ is the degree matrix which is diagonal, $A \in \mathbb{R}^{s\times s}$ is the adjacency matrix and $E \in \mathbb{R}^{s\times m}$ is the incidence matrix. Hence, $L \in \mathbb{R}^{s\times s}$. The entries of the adjacency matrix, A are denoted by a_{ij} and it is defined as follows:

$$a_{ij} = \begin{cases} 1 & \text{if there exists a link between } i \text{ and } j \\ 0 & \text{otherwise} \end{cases}$$

For an undirected graph, the adjacency matrix is symmetric. The columns of the adjacency matrix are denoted by $\boldsymbol{a}_j$. Let the vector $\boldsymbol{e}_i$ represent the standard basis vectors in $\mathbb{R}^s$ such that only the i^{th} component of the vector is 1 as shown below.

$$\boldsymbol{e}_i = \begin{bmatrix} 0 & \cdots & 1 & \cdots & 0 \end{bmatrix}^\top$$

Suppose $\mathcal{Q} = \begin{bmatrix} Q_1^\top & Q_2^\top & \ldots & Q_\ell^\top \end{bmatrix}^\top$ where $Q_1, Q_2, \ldots, Q_\ell$ are wide rectangular matrices. Then, we have

$$\| \mathcal{Q} \|_F^2 \equiv \sum_{i=1}^{\ell} \| Q_i \|_2^2 \tag{1}$$

where Eq. (1) follows from the linearity properties of trace.

Consider the multi-agent system described by a discrete-time nonlinear (autonomous) dynamical system as

$$\boldsymbol{x}^+ = F(\boldsymbol{x}) \tag{2}$$

where $\boldsymbol{x} \in \mathbb{R}^N$ is the overall state of the multi-agent agent, $\boldsymbol{x}^+$ is the successor value of the state and F is the discrete-time nonlinear transition mapping.

Suppose at any given time i, denote the overall system state as $\boldsymbol{x}_i \in \mathbb{R}^N$ such that

$$\boldsymbol{x}_i = \begin{bmatrix} \boldsymbol{x}_{i,1}^\top & \boldsymbol{x}_{i,2}^\top & \ldots & \boldsymbol{x}_{i,n}^\top \end{bmatrix}^\top$$

where the first part of the subscript indicates the time and the second element in the subscript indicates the agent for which the states belong to such that $\boldsymbol{x}_{i,\alpha} \in \mathbb{R}^{n_\alpha}$ for any agent $\alpha \in \{1, \ldots, n\}$ and $N := \sum_{\alpha=1}^n n_\alpha$.

A. DMD and EDMD

Next, we recall the construction of the approximate Koopman operator using dynamic mode decomposition (DMD) [15] and extended DMD (EDMD) methods [8], [30]. Let the system state time-series data from the experiment or simulation of the multi-agent system (2) is given by

$$\begin{aligned} X_p &= \begin{bmatrix} \boldsymbol{x}_1 & \boldsymbol{x}_2 & \cdots & \boldsymbol{x}_{k-2} & \boldsymbol{x}_{k-1} \end{bmatrix} \\ X_f &= \begin{bmatrix} \boldsymbol{x}_2 & \boldsymbol{x}_3 & \cdots & \boldsymbol{x}_{k-1} & \boldsymbol{x}_k \end{bmatrix} \end{aligned} \tag{3}$$

Let $\{\psi_1, \ldots, \psi_M\}$ be the choice of dictionary functions or observables where $\psi_i \in L_2(\mathbb{R}^N, \mathcal{B}, \mu)$ and $\psi_i : \mathbb{R}^N \to \mathbb{C}$. Define a vector valued observable function $\Psi : \mathbb{R}^N \to \mathbb{C}^M$ as follows.

$$\Psi(\boldsymbol{x}) := \begin{bmatrix} \psi_1(\boldsymbol{x}) & \psi_2(\boldsymbol{x}) & \cdots & \psi_M(\boldsymbol{x}) \end{bmatrix}^\top.$$

Then, we have the following least squares problem that minimizes the error

$$\min_{\mathcal{K}} \ \| Y_f - \mathcal{K} Y_p \|_F^2 \tag{4}$$

such that

$$\begin{aligned} Y_p &= \Psi(X_p) = [\Psi(\boldsymbol{x}_1), \Psi(\boldsymbol{x}_2), \cdots, \Psi(\boldsymbol{x}_{k-1})] \\ Y_f &= \Psi(X_f) = [\Psi(\boldsymbol{x}_2), \Psi(\boldsymbol{x}_3), \cdots, \Psi(\boldsymbol{x}_k)] \end{aligned}$$

where $\mathcal{K} \in \mathbb{R}^m \times \mathbb{R}^m$ is the finite dimensional approximation of the Koopman operator defined on the space of observables Hereafter, we refer to the finite dimensional approximation of the Koopman operator as the Koopman operator only. The solution to the optimization problem (4) can be obtained analytically and the approximate Koopman operator is given by

$$\mathcal{K} = Y_f {Y_p}^\dagger \tag{5}$$

where ${Y_p}^\dagger$ is the pseudo-inverse of Y_p. The original states can be obtained from the observables as a solution to the following minimization problem.

$$\min_{C} \ \| X_p - C Y_p \|_F^2 \tag{6}$$

where the solution to (6) can be analytically obtained as $C = X_p Y_p^\dagger$. Note that, DMD is a special case of EDMD algorithm with $\Psi(\boldsymbol{x}) = \boldsymbol{x}$ and $C = I$, where I denote the identity matrix.

B. Notations

Following notations are used in this paper. The number of nodes in a graph is denoted by s, number of agents are denoted by n, number of states at agent α are denoted by n_α such that $\sum_{\alpha=1}^n n_\alpha := N$, number of dictionary functions denoted by M, number of dictionary functions at agent α are denoted by m_α such that $\sum_{\alpha=1}^n m_\alpha := M$. The vectors $\boldsymbol{a}_j, \boldsymbol{e}_j$ respectively denote the j^{th} column of the adjacency matrix, A and a vector of standard basis in $\mathbb{R}^s$.

III. Distributed Koopman Operator Formulation

We present the main results of this work in this section. We begin with two necessary assumptions on the topology of the multi-agent system and its evolution. Eventually, we present the distributed Koopman operator computation and show that it is equivalent to the centralized Koopman operator computation.

Assumption 1 (Knowledge of Topology): The topology of the network on which the multi-agent system (2) evolves is known.

The following assumption leads to the identification of Koopman operator in a distributed manner.

Assumption 2 (Agent Evolution): For any given agent, the dynamic evolution of its states depend only on the states of the given agent and the states of its neighbors.

The time-series data for the multi-agent system (2) given in (3) corresponding to any agent $\alpha \in \{1, 2, \ldots, n\}$ is denoted by

$$\begin{aligned} X_{p,\alpha} &= \begin{bmatrix} \boldsymbol{x}_{1,\alpha} & \boldsymbol{x}_{2,\alpha} & \ldots & \boldsymbol{x}_{k-2,\alpha} & \boldsymbol{x}_{k-1,\alpha} \end{bmatrix} \\ X_{f,\alpha} &= \begin{bmatrix} \boldsymbol{x}_{2,\alpha} & \boldsymbol{x}_{3,\alpha} & \ldots & \boldsymbol{x}_{k-1,\alpha} & \boldsymbol{x}_{k,\alpha} \end{bmatrix} \end{aligned} \tag{7}$$

for every agent $\alpha \in \{1, 2, \ldots, n\}$ where the the first element of the subscript denotes the time snapshot information and the second element denote the agent information.

Recall from (5), $\mathcal{K}$ denotes the global Koopman operator for the system (2). We partition the global Koopman into the transition mapping associated with each agent as follows.

$$\mathcal{K} = \left[\begin{array}{c} \mathcal{K}_1 \\ \hline \mathcal{K}_2 \\ \hline \vdots \\ \hline \mathcal{K}_n \end{array}\right] = \left[\begin{array}{c|c|c|c} \mathcal{K}_{11} & \mathcal{K}_{12} & \cdots & \mathcal{K}_{1n} \\ \hline \mathcal{K}_{21} & \mathcal{K}_{22} & \cdots & \mathcal{K}_{2n} \\ \hline \vdots & \vdots & \ddots & \vdots \\ \hline \mathcal{K}_{n1} & \mathcal{K}_{n2} & \cdots & \mathcal{K}_{nn} \end{array}\right] \tag{8}$$

where each row(s) of $\mathcal{K}$, denoted by $\mathcal{K}_\alpha$ indicates the transition mapping associated to the agent α. Observe the Koopman operator $\mathcal{K}$ is a $n \times n$ block matrix where n are the number of agents.

In what follows, we formally establish an algorithm to identify the Koopman operator in a distributive manner.

A. Distributive EDMD for Multi-Agent Systems

Based on the Assumptions 1 and 2, it follows that the observable space can be defined for each agent as shown below. Let $\Psi_\alpha : \mathbb{R}^{n_\alpha} \to \mathbb{C}^{m_\alpha}$ be the vector-valued observable functions corresponding to the agent α. Define,

$$Y_{p,\alpha} := \Psi_\alpha(X_{p,\alpha}) = \begin{bmatrix} \Psi_\alpha(\boldsymbol{x}_{1,\alpha}) & \Psi_\alpha(\boldsymbol{x}_{2,\alpha}) & \cdots & \Psi_\alpha(\boldsymbol{x}_{k-2,\alpha}) & \Psi_\alpha(\boldsymbol{x}_{k-1,\alpha}) \end{bmatrix}$$

$$Y_{f,\alpha} := \Psi_\alpha(X_{f,\alpha}) = \begin{bmatrix} \Psi_\alpha(\boldsymbol{x}_{2,\alpha}) & \Psi_\alpha(\boldsymbol{x}_{3,\alpha}) & \cdots & \Psi_\alpha(\boldsymbol{x}_{k-1,\alpha}) & \Psi_\alpha(\boldsymbol{x}_{k,\alpha}) \end{bmatrix}$$

for every $\alpha \in \{1, 2, \ldots, n\}$. We now have $\mathcal{K}_{\alpha\alpha} \in \mathbb{R}^{m_\alpha \times m_\alpha}$, and $\mathcal{K}_{\alpha\beta} \in \mathbb{R}^{m_\alpha \times m_\beta}$.

Since the observables for each agent are identified only based on the agent state space data, the projections from the observable space to the state can be obtained by solving the following minimization problem.

$$\min_{C_\alpha} \parallel X_{p,\alpha} - C_\alpha Y_{p,\alpha} \parallel_F^2 \tag{9}$$

and the projection operator for the entire system can be obtained by, $C = \text{blkdiag}(C_1, C_2, \ldots, C_n)$.

Remark 3: In order to obtain the one time-step forwarded time-series data corresponding to agent α, we have,

$$\mathcal{K}_\alpha Y_p = \begin{bmatrix} \mathcal{K}_{\mathfrak{N}_\alpha} & \mathcal{K}_{\overline{\mathfrak{N}_\alpha}} \end{bmatrix} \begin{bmatrix} Y_{p,\mathfrak{N}_\alpha} \\ Y_{p,\overline{\mathfrak{N}_\alpha}} \end{bmatrix}$$

where $\mathfrak{N}_\alpha$ denotes the neighboring agents of agent α along with the agent itself and $\overline{\mathfrak{N}_\alpha}$ denote the non-neighboring agents of agents α such that $\mathfrak{N}_\alpha \cup \overline{\mathfrak{N}_\alpha}$ denote the set of agents $\{1, 2, \cdots, n\}$. Hence, $\mathcal{K}_{\mathfrak{N}_\alpha}$ and $\mathcal{K}_{\overline{\mathfrak{N}_\alpha}}$ respectively denote the (rectangular) matrices associated with the agent α, its neighbors and the non-neighbors of agent α. Similarly the time-series data, Y_p is partitioned according to the neighbors, denoted by $Y_{p,\mathfrak{N}_\alpha}$ and non-neighbors, denoted by $Y_{p,\overline{\mathfrak{N}_\alpha}}$.

Before presenting the distributed Koopman operator computation, we define the following transformation matrices, $T_{f,\alpha}, T_{p,\alpha}$ associated with each agent as follows.

$$\begin{aligned} T_{f,\alpha} &:= \text{blkdiag}(ee_1, ee_2, \ldots, ee_n) \\ T_{p,\alpha} &:= \text{blkdiag}(ae_1, ae_2, \ldots, ae_n) \end{aligned}$$

where

$$\begin{aligned} ee_i &= (\boldsymbol{e}_\alpha)_i \otimes I_{m_i} \quad \text{for } i \in \{1, 2, \ldots, n\} \\ ae_i &= (\boldsymbol{a}_\alpha + \boldsymbol{e}_\alpha)_i \otimes I_{m_i} \quad \text{for } i \in \{1, 2, \ldots, n\} \end{aligned}$$

where $(\cdot)_i$ denotes the i^{th} entry of the vector $(\cdot)$ and the matrices, $R_{p,\alpha}$ and $R_{f,\alpha}$ denote the transformation matrices that remove zero rows. In the following, we present the distributed Koopman operator computation and its corresponding workflow is shown in Fig. 1.

Algorithm 1 Distributed Koopman Operator Computation

Given an n agent network.

for $\alpha = 1,\ \alpha \leq n,\ \alpha = \alpha + 1$ **do**

1) Compute $Y_{f,\alpha}, Y_{p,\mathfrak{N}_\alpha}$ associated with agent α as

$$\begin{aligned} Y_{f,\alpha} &= R_{f,\alpha} T_{f,\alpha} Y_f \\ Y_{p,\mathfrak{N}_\alpha} &= R_{p,\alpha} T_{p,\alpha} Y_p \end{aligned}$$

2) Solve the minimization problem

$$\min_{\mathcal{K}_{\mathfrak{N}_\alpha}} \parallel Y_{f,\alpha} - \mathcal{K}_{\mathfrak{N}_\alpha} Y_{p,\mathfrak{N}_\alpha} \parallel_F^2$$

3) Compute $\hat{\mathcal{K}}_\alpha = \mathcal{K}_{\mathfrak{N}_\alpha} R_{p,\alpha}$

end

The distributed Koopman operator is given by

$$\mathcal{K}_D = \begin{bmatrix} \hat{\mathcal{K}}_1^\top & \hat{\mathcal{K}}_2^\top & \cdots & \hat{\mathcal{K}}_n^\top \end{bmatrix}^\top$$

Theorem 4: Under assumptions 1 and 2, the distributed approximate finite-dimensional Koopman operator (Algorithm 1) is equivalent to the approximate finite-dimensional Koopman operator computed using EDMD algorithm (4).

Proof. We first show the sufficiency condition and the necessity follows be reversing the steps. Consider the original

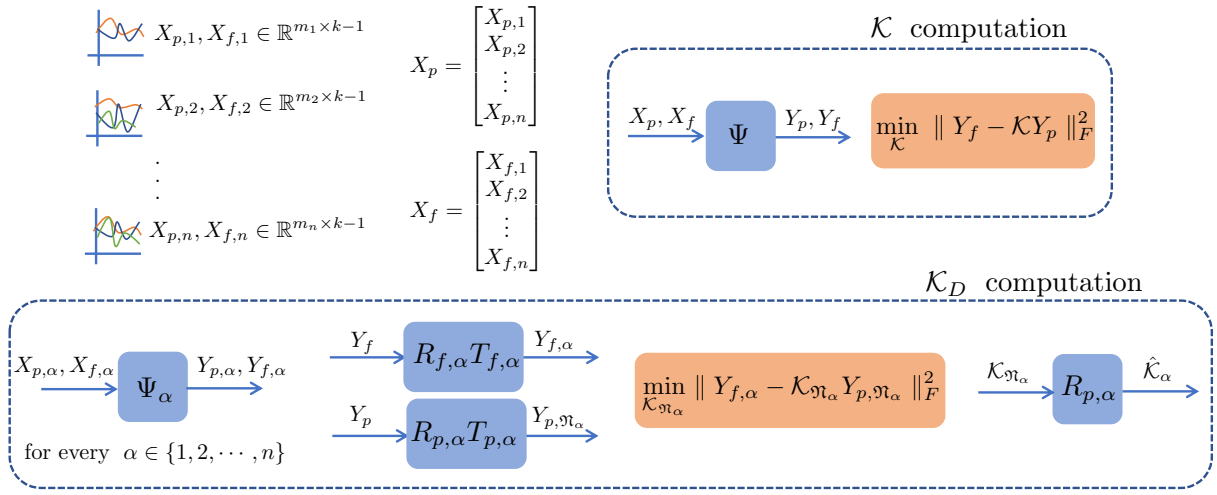


Fig. 1. Overview of the Koopman operator computation suing EDMD and the proposed distributed DMD.

EDMD learning problem given in Eq. (4) and rewrite it as associated with each agent as follows. Consider,

$$
\begin{aligned}
&\| Y_f - \mathcal{K} Y_p \|_F^2 \\
&= \left\| \begin{bmatrix} Y_{f,1} - \mathcal{K}_1 Y_p \\ Y_{f,2} - \mathcal{K}_2 Y_p \\ \vdots \\ Y_{f,n} - \mathcal{K}_n Y_p \end{bmatrix} \right\|_F^2 \\
&= \sum_{\alpha=1}^{n} \| Y_{f,\alpha} - \mathcal{K}_\alpha Y_p \|_F^2 \quad \text{(from Eq. (1))} \\
&= \sum_{\alpha=1}^{n} \| Y_{f,\alpha} - \mathcal{K}_{\mathfrak{N}_\alpha} Y_{p,\mathfrak{N}_\alpha} - \mathcal{K}_{\overline{\mathfrak{N}}_\alpha} Y_{p,\overline{\mathfrak{N}}_\alpha} \|_F^2 \quad \text{(from Remark 3)} \\
&= \sum_{\alpha=1}^{n} \| Y_{f,\alpha} - \mathcal{K}_{\mathfrak{N}_\alpha} Y_{p,\mathfrak{N}_\alpha} \|_F^2 + \| \mathcal{K}_{\overline{\mathfrak{N}}_\alpha} Y_{p,\overline{\mathfrak{N}}_\alpha} \|_F^2 \\
&\quad - 2 \operatorname{trace} \left((Y_{f,\alpha} - \mathcal{K}_{\mathfrak{N}_\alpha} Y_{p,\mathfrak{N}_\alpha})^\top \mathcal{K}_{\overline{\mathfrak{N}}_\alpha} Y_{p,\overline{\mathfrak{N}}_\alpha} \right) \\
&= \sum_{\alpha=1}^{n} \| Y_{f,\alpha} - \mathcal{K}_{\mathfrak{N}_\alpha} Y_{p^{\mathfrak{N}_\alpha}} \|_F^2 \quad \text{(under Assumption 2)}
\end{aligned}
$$

■

Corollary 5: The transition mappings corresponding to each agent α are obtained as a solution to the following minimization problem.

$$\min_{\mathcal{K}_{\mathfrak{N}_\alpha}} \| Y_{f,\alpha} - \mathcal{K}_{\mathfrak{N}_\alpha} Y_{p,\mathfrak{N}_\alpha} \|_F^2 \tag{10}$$

Then, the transition mapping corresponding to the agent α is given by

$$\hat{\mathcal{K}}_\alpha = \mathcal{K}_{\mathfrak{N}_\alpha} R_{p,\alpha}$$

Proof. Follows directly from Theorem 4. ■

Remark 6: The distributed Koopman, $\mathcal{K}_D$ is a block matrix and has the zero blocks corresponding to the zero entries of the Laplacian matrix L and hence the sparsity structure of $\mathcal{K}_D$ depends on the connectivity of the underlying multi-agent system. Furthermore, it is important to reiterate that $\mathcal{K}_D$ is not a symmetric matrix.

The ensuing result establishes that the distributive and centralized learning problems coincide when the network is fully connected.

Corollary 7: For a fully-connected network, the distributed learning problem and the centralized learning problem yields the same Koopman operator.

Proof. The proof follows by noticing that for any agent α, the corresponding non-neighbors set, $\overline{\mathfrak{N}_\alpha}$ is empty. ■ Although the above results were discussed for the EDMD case, under the assumption that $\Psi(x) = x$ and $C = I$, all the above results apply for DMD as well.

IV. Numerical Study

In this section, the proposed distributive Koopman operator computation is illustrated on two different dynamical systems, a network of oscillators and a power network along with comparisons to existing works such as DMD and sparse DMD.

A. Network of oscillators

The dynamical system associated with the network of oscillators is given by

$$\begin{bmatrix} \dot{\theta} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} O_n & I_n \\ -\gamma \mathcal{M}^{-1} L & -\mathcal{M}^{-1} D \end{bmatrix} \begin{bmatrix} \theta \\ \dot{\theta} \end{bmatrix} \tag{11}$$

where $\theta \in \mathbb{R}^n$ and $\dot{\theta} \in \mathbb{R}^n$ represents the angle and angular speed of the linear oscillator respectively with n being the number of agents. The matrices $\mathcal{M} \in \mathbb{R}^{n \times n}$, $D \in \mathbb{R}^{n \times n}$ are diagonal and the respective entries denote the inertia and damping of each oscillator. The coupling between the agents is captured by the Laplacian L and β represents the strength of the interconnections. The matrices O_n and I_n respectively denote the zero and identity matrices of size n. The set of equilibrium points of the system (11) is the synchronization manifold.

In a network of few oscillators (around 10 or less), the DMD and the distributive DMD yield no differences and in particular results in the same Koopman operator structure, same eigenspectrum, and same predictive performance. The superiority of the proposed distributive approach to identify the finite-dimensional Koopman becomes noticeable as the number of agents increases. Hence, we consider a network with 100 oscillators and demonstrate the efficiency of the proposed distributive approach when the network is highly sparse.

To measure the sparsity of the network, we evaluate the metric network density, which is defined as shown below.

$$\text{Network density} = \frac{\text{total number of edges}}{\text{total possible number of edges}}$$

Essentially, the network density is a quantity that always lies in the range $(0, 1]$, where the small values of the network density indicate a sparse network and for a fully-connected network, it takes the value of 1.

Case study I - Varying number of neighbors: We consider a small-world network with 100 oscillators and gradually increase the number of interconnections such that for any oscillator, the number of neighboring oscillators varies from 10 (sparse network) to 99 (fully connected network), as shown in Fig. 2 (top plot). Network parameters such as inertia, damping are chosen randomly and then kept constant for all the different sets of simulations. The time-series data for each scenario is generated by simulating each system for 30s with 0.05s sampling time with 5 different randomly chosen initial conditions.

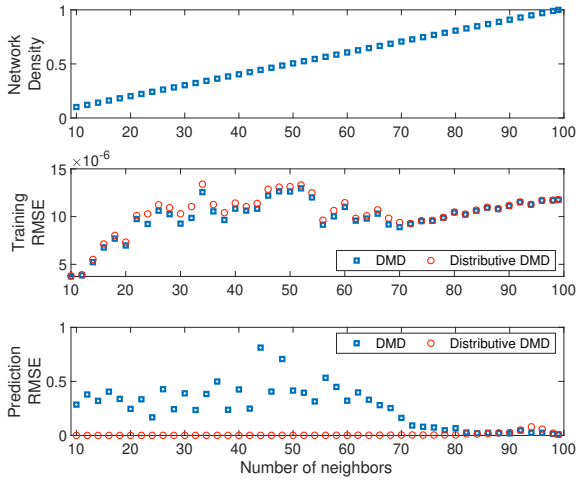


Fig. 2. For a small-world network with 100 oscillators, comparison of the training and prediction RMSE using DMD and distributed DMD while the network density is varied.

For each choice of the network density, both DMD and distributed DMD algorithm is used to compute the Koopman operator and it is observed that the training RMSE (see Fig. 2 middle plot) for both DMD and distributed DMD remain small (in the order of 10^{-6}). However, distributed DMD performs better in predicting the future evolution of the states (prediction window was 30s for each scenario), especially when the network is sparsely connected as can be seen in the bottom plot of Fig. 2.

Case study II - Varying number of agents: In the second set of simulations, we consider small-world networks, such that for each scenario each agent had 10 neighbors. However, the number of agents in the network are increased from 20 to 100, such that the network density decreases gradually, that is, the network is made more sparse, as shown in Fig. 3 (top plot). Similar to the case study on varying the number of neighbors, the training RMSE for both DMD and distributed DMD are very small (see Fig. 3 middle plot). But as the sparsity in the network is increased, the distributed DMD outperformed normal DMD, as far as prediction performance is concerned (see Fig. 3 bottom plot). The results are consistent with the previous case in the sense that when the network sparsity is high the distributed DMD performs much better compared to normal DMD for predicting the future evolution of the states.

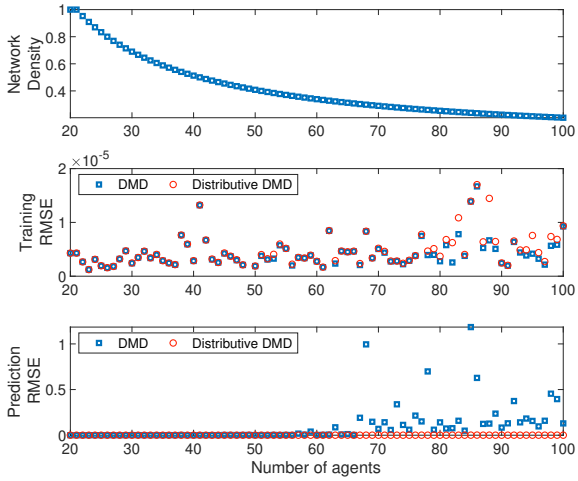


Fig. 3. For a small-world network with 10 neighboring oscillators for each oscillator, comparison of the training and prediction RMSE using DMD and distributed DMD while increasing the number of oscillators.

The analysis shows that while the network is sparse, the DMD approach fails to identify the exact dynamics and network structure of the underlying system. In particular, it fails to capture the sparse topology and this leads to a degradation in prediction. However, the distributed DMD approach that relies on the knowledge of the underlying topology captures the correct connectivity and thus has a better prediction performance. In particular, since centralized DMD does not incorporate the knowledge of the topology of the underlying network it essentially tries to find N^2 (assuming that the Koopman operator $\mathcal{K} \in \mathbb{R}^{N \times N}$) numbers that best fit the data. As such, it has no hard constraint to make any of its N^2 element zero. However, our distributed approach forcefully makes some of the elements of the Koopman operator zero

(depending on the topology) and then solves the minimization problem.

DMD, Sparse DMD and distributed DMD: Sparse DMD, introduced in [28] addresses the problem of identifying a Koopman operator that is sparse by imposing additional sparsity promoting cost in (4) but the resultant minimization problem doesn't have an analytical solution and is thus computationally expensive. For comparing the proposed distributed DMD algorithm with sparsity promoting DMD, we considered a small-world network of 35 oscillators, such that any oscillator has a maximum of 3 neighbors. With a weight of 0.01 for the sparsity promoting cost in sparse DMD (see [28] for more details), it is observed that the sparse DMD and the distributed DMD result in Koopman operator that is sparse whereas DMD is only diagonally dominant (see Fig. 4 (a)). Moreover, the eigenspectrum coincides for distributive DMD and sparse DMD, but not for DMD as shown in Fig. 4 (b). However, leveraging the learned models for prediction it can be seen that the prediction RMSE for distributed DMD is orders of magnitude smaller than sparse DMD and DMD methods (refer Fig. 4 (c)).

Advantage of Distributed DMD: To address the advantages of the proposed distributed algorithm and show the challenges associated with existing works, we consider four different sparse small-world networks with $15, 35, 60, 100$ oscillators. The network densities of the different networks, training RMSE, prediction RMSE, and the computational time for each network applying each method are shown in Table I. The DMD, sparse DMD, and distributed DMD have been performed on the computer with configuration: 2.6 GHz Intel Core i7 processor with 16GB of RAM. It is clear that the training RMSE for every sparse small-world network is small under DMD, sparse DMD, and distributed DMD. However, the prediction RMSE is orders of magnitude higher for both DMD and sparse DMD compared to distributed DMD. Moreover, observing the computational aspects of each of the models indicate that the DMD and distributed DMD is quite fast even for a 100 oscillator network (200 states) whereas the computational complexity of sparse DMD increases rapidly with an increase in the number of states.

Table I. Training and Predictive comparison of different approaches for different sparse small-world networks. The weight for sparsity promoting cost is chosen to be 0.01 for all the networks. Color coding in the table indicates that smaller values are shown in light green, intermediate values in yellow and large values in red.

	Number of states	Network density	Training RMSE	Prediction RMSE	Computation time [s]
DMD	30	0.1429	6.99E-05	0.00013	0.001
	70	0.0588	8.98E-05	0.27	0.001
	120	0.0399	7.52E-05	0.91	0.003
	200	0.0202	1.51E-04	0.65	0.01
Sparse DMD	30	0.1429	7.58E-05	1.21E-03	21.5
	70	0.0588	9.97E-05	0.04	238.85
	120	0.0399	6.53E-04	0.31	1072.78
	200	0.0202	6.27E-03	0.68	4503.23
Distributed DMD	30	0.1429	7.11E-05	2.01E-05	0.01
	70	0.0588	9.05E-05	1.17E-05	0.01
	120	0.0399	7.95E-05	1.75E-05	0.23
	200	0.0202	1.60E-04	3.09E-05	0.82

It is important to mention here that for sparse DMD, since the level of the sparsity of the Koopman operator depends on the choice of the weight for the sparsity promoting cost, this weight needs to be refined to obtain sparsity seen in the case of distributed DMD and especially for large-scale networks this is computationally expensive. Overall, the distributive DMD combines the computational benefits of DMD and sparsity of sparse DMD, while ensuring scalability and minimal prediction errors.

B. IEEE 68 Bus System

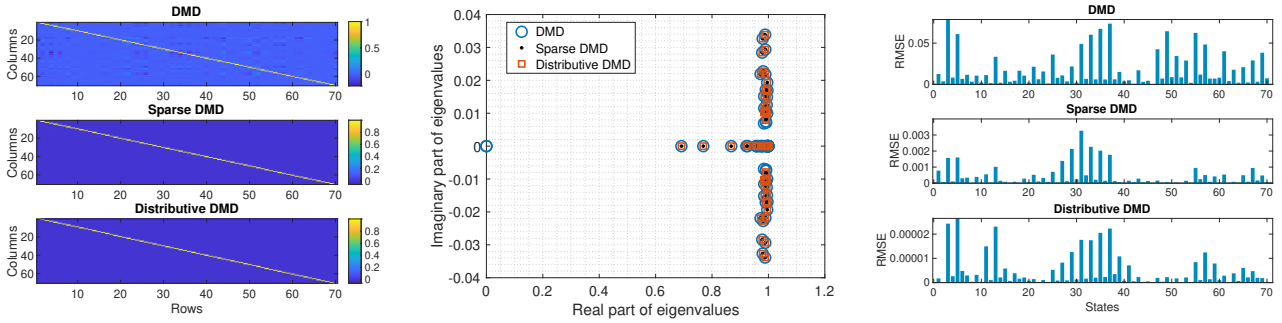
In this subsection, we consider a 68-bus, 16-machine, 5-area system which is a reduced order equivalent of the inter-connected New England test system (NETS) and New York power system (NYPS), with five geographical regions out of which NETS and NYPS are represented by a group of generators whereas, the power import from each of the three other neighboring areas are approximated by equivalent generator models. The one-line diagram of the 68 bus system is shown in Fig. 5 on the left and its equivalent multi-agent system with 5 agents is shown on the right.

Time-series data for this study is generated using GridSTAGE by considering random load changes in the 68 bus network. GridSTAGE is introduced in [31] which is a spatio-temporal adversarial data generation tool that emulates high frequency PMU and SCADA measurements. PMU measurements such as frequency and voltage magnitudes are considered for training DMD and distributed DMD models (based on the proposed distributed Algorithm 1).

The trained DMD and distributed DMD models are applied to predict the system states starting from an initial condition corresponding to a disturbance in the network. The dominant eigenvalues of the Koopman operators using DMD and distributed DMD are shown in Fig. 6 (a). The prediction RMSE for frequency states and the voltage magnitude states are very small and are close to each other for both DMD and distributed DMD approaches (see Fig. 6 (b)). In particular, prediction performance of both DMD and distributed DMD models are very similar as can be seen in Figs. 6 (c) and (d).

V. Conclusion

This work proposes a distributed algorithm to compute the Koopman operator from time-series data under the knowledge of the underlying system's topology. This results in learning the evolution of each agent as a function of itself and its neighbors and the global Koopman operator is obtained by stitching the individual Koopman operators. The main advantage of the proposed framework is the fact that the algorithm scales easily for large-scale networks, especially for sparse networks, while maintaining model accuracy all through. Moreover, simulation studies on a network of oscillators and the IEEE 68 bus system show that the computational time to obtain the Koopman operator using DMD and distributed DMD is in the same range. However, the evaluation of the DMD and distributed DMD model for predictions on a sparse network with 200



(a) Sparsity structure of the Koopman operator (b) Eigenspectrum of the Koopman operator (c) Prediction RMSE starting from a random initial condition

Fig. 4. Comparison of the distributed DMD with DMD, and sparse DMD for a sparse small-world network with 35 oscillators and network density of 0.0588.

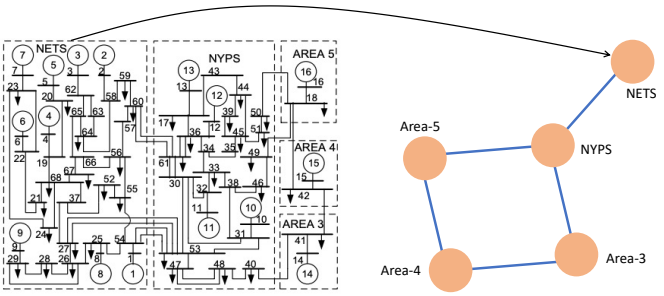


Fig. 5. New England IEEE 68 bus system, a 16-machine, 5-area network depicted as a multi-agent system where each agent corresponds to an area in the power network. Figure also shows the connection between the NETS part of the system and its agent representation on the right. Similar interpretation is made for other areas and agents.

states reveal that the prediction RMSE is orders of magnitude higher for DMD when compared to the distributed DMD.

References

[1] H. K. Khalil and J. W. Grizzle, *Nonlinear systems*. Prentice hall Upper Saddle River, NJ, 2002, vol. 3.

[2] B. O. Koopman, "Hamiltonian systems and transformation in Hilbert space," *Proceedings of the National Academy of Sciences*, vol. 17, no. 5, pp. 315–318, 1931.

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

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

[5] A. Lasota and M. C. Mackey, *Chaos, fractals, and noise: stochastic aspects of dynamics*. Springer Science & Business Media, 2013, vol. 97.

[6] J. H. Tu, C. W. Rowley, D. M. Luchtenburg, S. L. Brunton, and J. N. Kutz, "On dynamic mode decomposition: Theory and applications," *arXiv preprint arXiv:1312.0041*, 2013.

[7] J. N. Kutz, S. L. Brunton, B. W. Brunton, and J. L. Proctor, *Dynamic mode decomposition: data-driven modeling of complex systems*. SIAM, 2016.

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

[9] Q. Li, F. Dietrich, E. M. Bollt, and I. G. Kevrekidis, "Extended dynamic mode decomposition with dictionary learning: A data-driven adaptive spectral decomposition of the Koopman operator," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 27, no. 10, p. 103111, 2017.

[10] H. Arbabi and I. Mezic, "Ergodic theory, dynamic mode decomposition, and computation of spectral properties of the Koopman operator," *SIAM Journal on Applied Dynamical Systems*, vol. 16, no. 4, pp. 2096–2126, 2017.

[11] B. Huang and U. Vaidya, "Data-driven approximation of transfer operators: Naturally structured dynamic mode decomposition," in *2018 Annual American Control Conference (ACC)*. IEEE, 2018, pp. 5659–5664.

[12] E. Yeung, S. Kundu, and N. Hodas, "Learning deep neural network representations for koopman operators of nonlinear dynamical systems," in *2019 American Control Conference (ACC)*. IEEE, 2019, pp. 4832–4839.

[13] B. Lusch, J. N. Kutz, and S. L. Brunton, "Deep learning for universal linear embeddings of nonlinear dynamics," *arXiv preprint arXiv:1712.09707*, 2017.

[14] N. Takeishi, Y. Kawahara, and T. Yairi, "Learning Koopman invariant subspaces for dynamic mode decomposition," in *Advances in Neural Information Processing Systems*, 2017, pp. 1130–1140.

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

[16] S. Sinha, B. Huang, and U. Vaidya, "On robust computation of koopman operator and prediction in random dynamical systems," *Journal of Nonlinear Science*, pp. 1–34, 2019.

[17] U. Vaidya, "Observability gramian for nonlinear systems," in *Decision and Control, 2007 46th IEEE Conference on*. IEEE, 2007, pp. 3357–3362.

[18] A. Surana and A. Banaszuk, "Linear observer synthesis for nonlinear systems using Koopman operator framework," *IFAC-PapersOnLine*, vol. 49, no. 18, pp. 716–723, 2016.

[19] S. L. Brunton, B. W. Brunton, J. L. Proctor, and J. N. Kutz, "Koopman invariant subspaces and finite linear representations of nonlinear dynamical systems for control," *PloS one*, vol. 11, no. 2, p. e0150171, 2016.

[20] B. Huang, X. Ma, and U. Vaidya, "Feedback Stabilization Using Koopman Operator," *arXiv preprint arXiv:1810.00089*, 2018.

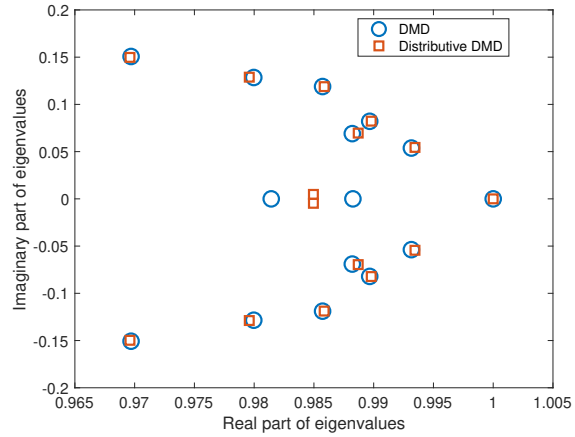
[21] M. Korda and I. Mezić, "Koopman model predictive control of nonlinear dynamical systems," in *The Koopman Operator in Systems and Control*. Springer, 2020, pp. 235–255.

[22] S. Sinha and U. Vaidya, "Data-driven approach for inferencing causality and network topology," in *2018 Annual American Control Conference (ACC)*. IEEE, 2018, pp. 436–441.

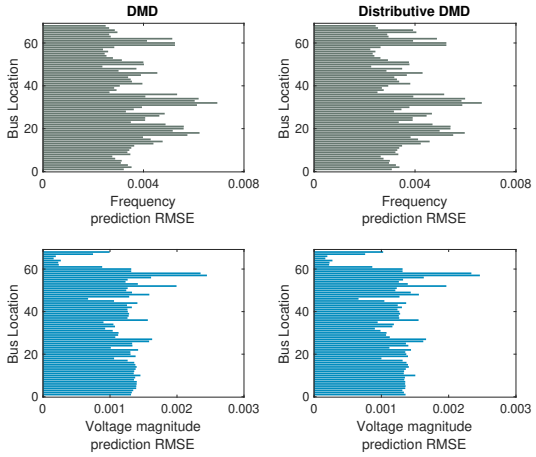
[23] ——, "On data-driven computation of information transfer for causal inference in discrete-time dynamical systems," *Journal of Nonlinear Science*, pp. 1–26, 2020.

[24] ——, "Formalism for information transfer in dynamical network," in *2015 54th IEEE Conference on Decision and Control (CDC)*. IEEE, 2015, pp. 5731–5736.

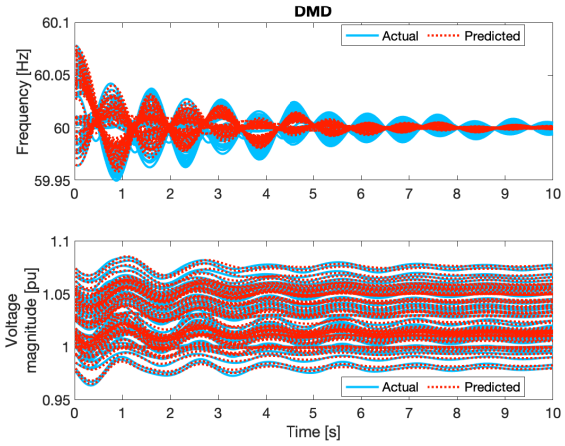
[25] ——, "Causality preserving information transfer measure for control dynamical system," in *2016 IEEE 55th Conference on Decision and Control (CDC)*. IEEE, 2016, pp. 7329–7334.



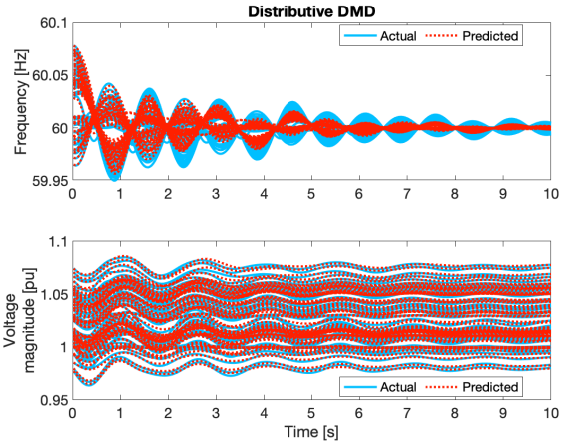
(a) Dominant eigenvalues whose absolute value is greater than 0.98



(b) Prediction RMSE at each bus



(c) The frequency and voltage magnitude predictions using DMD.



(d) The frequency and voltage magnitude predictions using distributive DMD

Fig. 6. Illustration of the proposed approach on the 5-area, IEEE 68 bus system.

[26] ——, "On information transfer in discrete dynamical systems," in *2017 Indian Control Conference (ICC)*. IEEE, 2017, pp. 303–308.

[27] N. Boddupalli, A. Hasnain, S. P. Nandanoori, and E. Yeung, "Koopman operators for generalized persistence of excitation conditions for nonlinear systems," in *2019 IEEE 58th Conference on Decision and Control (CDC)*. IEEE, 2019, pp. 8106–8111.

[28] M. R. Jovanovic, P. J. Schmid, and J. Nichols, "Low-rank and sparse dynamic mode decomposition," *Center for Turbulence Research Annual Research Briefs*, vol. 2012, pp. 139–152, 2012.

[29] Y. Li, H. He, J. Wu, D. Katabi, and A. Torralba, "Learning compositional koopman operators for model-based control," *arXiv preprint arXiv:1910.08264*, 2019.

[30] S. P. Nandanoori, S. Sinha, and E. Yeung, "Data-driven operator theoretic methods for global phase space learning," in *2020 American Control Conference (ACC)*. IEEE, 2020, pp. 4551–4557.

[31] S. P. Nandanoori, S. Kundu, S. Pal, K. Agarwal, and S. Choudhury, "Model-agnostic algorithm for real-time attack identification in power grid using koopman modes," in *2020 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*. IEEE, 2020, pp. 1–6.