World Academy of Science, Engineering and Technology
International Journal of Information and Communication Engineering
Vol:16, No:10, 2022

Model-Free Distributed Control of Dynamical Systems

Javad Khazaei, Rick S. Blum

Abstract—Distributed control is an efficient and flexible approach for coordination of multi-agent systems. One of the main challenges in designing a distributed controller is identifying the governing dynamics of the dynamical systems. Data-driven system identification is currently undergoing a revolution. With the availability of high-fidelity measurements and historical data, model-free identification of dynamical systems can facilitate the control design without tedious modeling of high-dimensional and/or nonlinear systems. This paper develops a distributed control design using consensus theory for linear and nonlinear dynamical systems using sparse identification of system dynamics. Compared with existing consensus designs that heavily rely on knowing the detailed system dynamics, the proposed model-free design can accurately capture the dynamics of the system with available measurements and input data and provide guaranteed performance in consensus and tracking problems. Heterogeneous damped oscillators are chosen as examples of dynamical system for validation purposes.

Keywords— Consensus tracking, distributed control, model-free control, sparse identification of dynamical systems.

I. Introduction

DATA-DRIVEN modeling of dynamical systems has recently been revolutionized with the advances on machine learning approaches and unprecedented availability of high-resolution data from historical records. Several approaches have been introduced for capturing the dynamics of complex systems including: (i) dynamic mode decomposition [1], [2], dynamic mode decomposition with control [3], which heavily relies on a linear dynamics assumption but can handle high-dimensional data, (ii) Koopman operator with control [4], [5] that connects dynamic mode decomposition to nonlinear dynamics through the Koopman operator, (iii) genetic programming which constructs categories of candidate nonlinear functions for the rate of change of state variables in time [6]. A model is then selected as a Pareto optimal solution that provides a balanced between model complexity and predictive accuracy. (iv) Recently, an approach was developed to automatically select from several candidate terms those terms which are most suitable to describe the dynamics. This method is called sparse identification of system dynamics (SINDY), and uses a sparse regression technique from machine learning to identify dominant dynamics of candidate functions, and has shown promise in accurately modeling the unknown dynamics of complex systems [7], [8]. One of the main advantages of SINDY for control purposes is the sparsity technique that reduces the training time and heavy reliance of neural-network-based approaches for identification and control which can be hard to interpret. The proposed approach is fully interpretable based on classical control theories. The application of SINDY for capturing input-output models that are appropriate for control design purposes was reported in [8], [9]. It was shown in [8] that sparse identification can capture dynamics of feedback control systems in nonlinear dynamics and its application to model-predictive control of aircraft dynamics was reported in [9]. While these studies show the significant improvement in control design of unknown dynamics, the reported studies mainly focused on centralized control approaches that might not be practical for large-scale systems with distributed complex dynamics, i.e., transportation systems, power grids, buildings.

Javad Khazaei is an assistant professor and Rick S. Blum is a professor in the Electrical and Computer Engineering (ECE) department at Lehigh University, Bethlehem, PA, United States (e-mail: khazaei@Lehigh.edu, & rblum@lehigh.edu).

Distributed control in this case has a few advantages over conventional centralized approaches: First, system security, reliability, and scalability are improved [10], as no single point of failure (central unit) exists. Second, computational efforts are divided to many nodes instead of being performed all at the central unit as in centralized mechanisms [10]. However, the application of model-free data-driven approaches for distributed control of large-scale complex systems has not been reported yet.

The goal of this paper is to investigate the application of a sparse identification approach for distributed control design of complex dynamics. Using the sparse regression technique, input-output dynamics of the unknown heterogeneous dynamical systems will be predicted. The learned dynamics can then be used to design distributed cooperative controllers that minimize the error between the desired and actual states. The tracking control of dynamical systems using the sparse identification technique will also be investigated. Contributions of the paper are listed as:

- Designing a distributed controller with minimum communication requirements for damped oscillators using only measurements
- Providing guaranteed stability of the designed consensus tracking control problem
- Accurately identifying the dynamics of damped oscillators using sparse identification technique

The rest of the paper is organized as follows: Section II formulates the sparse identification problem and Section III includes the proposed distributed control design. Numerical results are included in Section IV and Section V concludes the paper.

II. Model-Free Identification of Dynamical Systems

A robust approach in identifying the governing equations of nonlinear/linear systems is to construct families of candidate functions for the rate of change of state variables in time. Among all candidate functions, since most dynamical systems have few nonlinear terms in the dynamics, sparsity promoting techniques can be used to identify the candidate functions with most impact in forming the system dynamics using available measurements. This method is called sparse identification of nonlinear dynamics (SINDY), which was originally proposed in [7]. SINDY combines symbolic regression and sparse representation to come up with the dynamics of the system. Symbolic regression is a machine learning approach for determining a function relating the input to output using available data. In this research, SINDY will be used to identify governing dynamics of a dynamical system for distributed control design purposes. In the following, an overview of sparse identification technique is included.

The sparse identification relies on the fact that many dynamical systems of the form $\dot{x} = f(x, u)$ have relatively few terms in the right hand side of their governing equations. We assume that the actual dynamics of a system is represented by:

$$\frac{d}{dt}\mathbf{x}(t) = \dot{\mathbf{x}}(t) = f(\mathbf{x}(t), \mathbf{u}(t)) \tag{1}$$

where $\mathbf{x}(t) = [x_1(t)\ x_2(t)\ \ldots\ x_n(t)] \in \mathbb{R}^n$ is a vector of states and $\mathbf{u}(t) \in \mathbb{R}^n$ is a vector of controllable inputs. In regression problems, only a few terms are important and sparse feature selection can be used to identify the most dominant terms representing the dynamics.

To identify the governing equations of the system in (1), a time-history of the state vector $\mathbf{x}(t)$, input $\mathbf{u}(t)$, and $\dot{\mathbf{x}}(t)$ is required. In most practical systems, only $\mathbf{x}(t)$ and $\mathbf{u}(t)$ are available and $\dot{\mathbf{x}}(t)$ needs to be estimated from $\dot{\mathbf{x}}(t)$. If the measurement data is sampled at m intervals $t_1,\ t_2,\ \ldots,\ t_m$ and measurements are arranged into a matrix $\mathbf{X}$,

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}^T(t_1) \\ \mathbf{x}^T(t_2) \\ \vdots \\ \mathbf{x}^T(t_m) \end{bmatrix} = \begin{bmatrix} x_1(t_1) & x_2(t_1) & \ldots & x_n(t_1) \\ x_1(t_2) & x_2(t_2) & \ldots & x_n(t_2) \\ \vdots & \vdots & \ddots & \vdots \\ x_1(t_m) & x_2(t_m) & \ldots & x_n(t_m) \end{bmatrix} \tag{2}$$

and inputs for t_m samples are written into a matrix $\mathbf{U}$,

$$\mathbf{U} = \begin{bmatrix} \mathbf{u}^T(t_1) \\ \mathbf{u}^T(t_2) \\ \vdots \\ \mathbf{u}^T(t_m) \end{bmatrix} = \begin{bmatrix} u_1(t_1) & u_2(t_1) & \ldots & u_n(t_1) \\ u_1(t_2) & u_2(t_2) & \ldots & u_n(t_2) \\ \vdots & \vdots & \ddots & \vdots \\ u_1(t_m) & u_2(t_m) & \ldots & u_n(t_m) \end{bmatrix} \tag{3}$$

the measurements for derivatives can be approximated numerically from $\mathbf{X}$ following the procedure in the next section.

A. Savitzky-Golay Filtering

In [11], Savitzky-Golay developed a filter that is used to smooth out a noisy signal. This is necessary before estimating the derivatives of $\mathbf{X}$. Savitzky-Golay filters are also called digital smoothing polynomial filters or least-square smoothing filters. The main advantage of these filters compared to other filters (i.e., finite impulse response (FIR) filters) is their ability to minimize the least-squares errors in fitting a polynomial to frames of noisy data. The basic idea behind Savitzky-Golay filter is that having a group of $2M+1$ samples of a signal $x[n]$, centered at $n = 0$, the coefficients of a polynomial $p(n)$ that minimizes the mean-squared approximation error for the group of samples are obtained [12]. The polynomial is defined as:

$$p(n) = \sum_{k=0}^{N} a_k n^k \tag{4}$$

and the coefficients a_k are obtained to minimize the mean-squared approximation error (MSE) expressed by [12]:

$$\text{MSE} = \sum_{n=-M}^{M} (p(n) - x[n])^2 \tag{5}$$

where M is denoted as the half-width of the approximation interval. Savitzky and Golay showed in [11] that during each interval, the output obtained by sampling the fitted polynomial is equivalent to a fixed linear combination of the local set of input samples. This observation simplified the smoothing process by the fact the the output samples can be computed by a discrete convolution sum of the form [12]:

$$y[n] = \sum_{m=-M}^{M} h[m]\ x[n-m] \tag{6}$$

instead of differentiating (5) with respect to each of $N+1$ unknown coefficients of the polynomial and setting the corresponding derivative equal to zero. In this paper, the existing Savitzky-Golay filter function of MATLAB is used to smooth out the measurement samples $\mathbf{X}$ before estimating the derivatives.

B. Estimating the Derivatives, $\dot{\mathbf{X}}$

Difference approximations are used to numerically solve for the solution of ordinary and partial differential equations. Considering a smooth function in the neighborhood of point x, the derivatives can be approximated using Taylor series expansion at specified mesh points. Since the central difference approximation is more accurate for smooth functions, it is used in our paper. In this case, $\dot{\mathbf{X}}$ can be approximated by [13]:

$$\dot{\mathbf{X}} \approx \frac{\mathbf{X_f}(i+1) - \mathbf{X_f}(i-1)}{2h} \tag{7}$$

where $\mathbf{X_f}(i+1)$ is the filtered data at sample $i+1$ and h is the mesh spacing, which is considered the same as the sampling time of the simulation in this study, i.e., $5e^{-5}$ seconds.

C. Sparse Identification of System Dynamics

Having calculated $\dot{\mathbf{X}}$, the library of candidate functions will be constructed as linear and nonlinear functions of the columns of $\mathbf{X}$ and $\mathbf{U}$. A typical choice of candidate functions include polynomials and trigonometric functions for nonlinear systems such as (8).

In (8), $\mathbf{P_2}(\mathbf{X}, \mathbf{U})$ and $\mathbf{P_3}(\mathbf{X}, \mathbf{U})$ denote nonlinear combination of second- and third-order polynomials of $\mathbf{X}$ and

$$\Theta(\mathbf{X},\mathbf{U}) = \begin{bmatrix} | & | & | & | & | & & | & | & | & \\ 1 & \mathbf{X} & \mathbf{U} & \mathbf{P_2}(\mathbf{X},\mathbf{U}) & \mathbf{P_3}(\mathbf{X},\mathbf{U}) & \dots & \sin(\mathbf{X},\mathbf{U}) & \cos(\mathbf{X},\mathbf{U}) & \sin(\mathbf{2}(\mathbf{X},\mathbf{U})) & \dots \\ | & | & | & | & | & & | & | & | & \end{bmatrix} \tag{8}$$

$$\mathbf{P_2}(\mathbf{X},\mathbf{U}) = \begin{bmatrix} x_1^2(t_1) & x_1(t_1)x_2(t_1) & \dots & x_2^2(t_1) & \vdots & x_1(t_1)u_1(t_1) & \dots & u_1^2(t_1) & \dots \\ x_1^2(t_2) & x_1(t_2)x_2(t_2) & \dots & x_2^2(t_2) & \vdots & x_1(t_2)u_2(t_2) & \dots & u_2^2(t_2) & \dots \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \dots & \\ x_1^2(t_m) & x_1(t_m)x_2(t_m) & \dots & x_2^2(t_m) & \vdots & x_1(t_m)u_1(t_m) & \dots & u_1^2(t_m) & \dots \end{bmatrix} \tag{9}$$

$\mathbf{U}$ columns, respectively. For example, for the second-order polynomial, the candidate function $\mathbf{P_2}(\mathbf{X},\mathbf{U})$ is expressed in (9) as:

The sparse coefficients of the matrix $\mathbf{\Gamma}$ can be solved using the following equation [8]:

$$\dot{\mathbf{X}} = \Theta(\mathbf{X},\mathbf{U})\mathbf{\Gamma}, \tag{10}$$

where each column of Γ represents a sparse vector of coefficients identifying which terms are active. The coefficients of Γ can be found using the sparse regression formulation presented in **Algorithm 1**. If the intent is to identify the signal $\mathbf{U}$ for feedback control, i.e., $\mathbf{U} = G(s)\mathbf{X}$, where $G(s)$ is the transfer function of the controller, the matrix of inputs can be identified using [8]:

$$\mathbf{U} = \Theta(\mathbf{X})\mathbf{\Gamma_u} \tag{11}$$

where $\Theta(\mathbf{X})$ is the matrix of candidate functions and the terms corresponding to $\mathbf{U}$ have been removed from $\Theta(\mathbf{X})$, i.e., as in (12). $\mathbf{\Gamma_u}$ can be found using the sparse regression algorithm similar to $\mathbf{\Gamma}$.

In summary, (10) predicts the dynamics of the system using available measurements and then the predicted dynamics can be used for control design purposes, which will be discussed in the next section. An example of a two-dimensional controlled damped harmonic oscillator with linear dynamics is considered to validate the effectiveness of the sparse regression algorithm in identifying the dynamics. The dynamic system is represented by equaation (13).

After learning the dynamics, the system with these dynamics was first run for 25 seconds with a random input shown in the third subplot in Fig. 1 and the model was trained for this input. Dynamic response of x_1 and x_2 in response to this input is depicted in the first two subplots in Fig. 1 and compared to the actual model in (15), confirming a perfect prediction of the regression model. The input then was changed from 25 to 50 seconds to a completely different type (sinusoidal) that the model was not trained for, as it can be observed, the sparse identification results give accurate prediction of system dynamics and inputs for this example.

Algorithm 1 Sparse Regression Algorithm

Input: Measurements $\mathbf{X}, \mathbf{U}$
Input: Estimated derivatives$\dot{\mathbf{X}}$

```
procedure LEAST-SQUARE
Γ = Θ\Ẋ (least-square solution)
    for k = 1 : 10 do (number of iterations)
        Set λ (sparcification knob)
        |Γ| < λ ⟶ ind_small
        Γ(ind_small) ⟶ 0
        for k = 1 : n do (n dimension of state X)
            ind_big ≠ ind_small(:, k)
            Γ(ind_big, k) = Θ(:, ind_big)\Ẋ(:, k)
        end for
    end for
```

Output: sparse matrix Γ

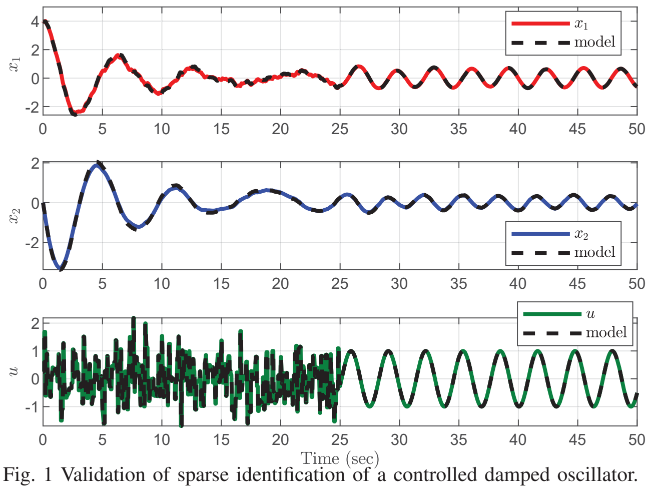

Fig. 1 Validation of sparse identification of a controlled damped oscillator.

$$\Theta(\mathbf{X}) = \begin{bmatrix} | & | & | & | & & | & | & | & \\ 1 & \mathbf{X} & \mathbf{P_2}(\mathbf{X}) & \mathbf{P_3}(\mathbf{X}) & \dots & \sin(\mathbf{X}) & \cos(\mathbf{X}) & \sin(\mathbf{2X}) & \dots \\ | & | & | & | & & | & | & | & \end{bmatrix} \tag{12}$$

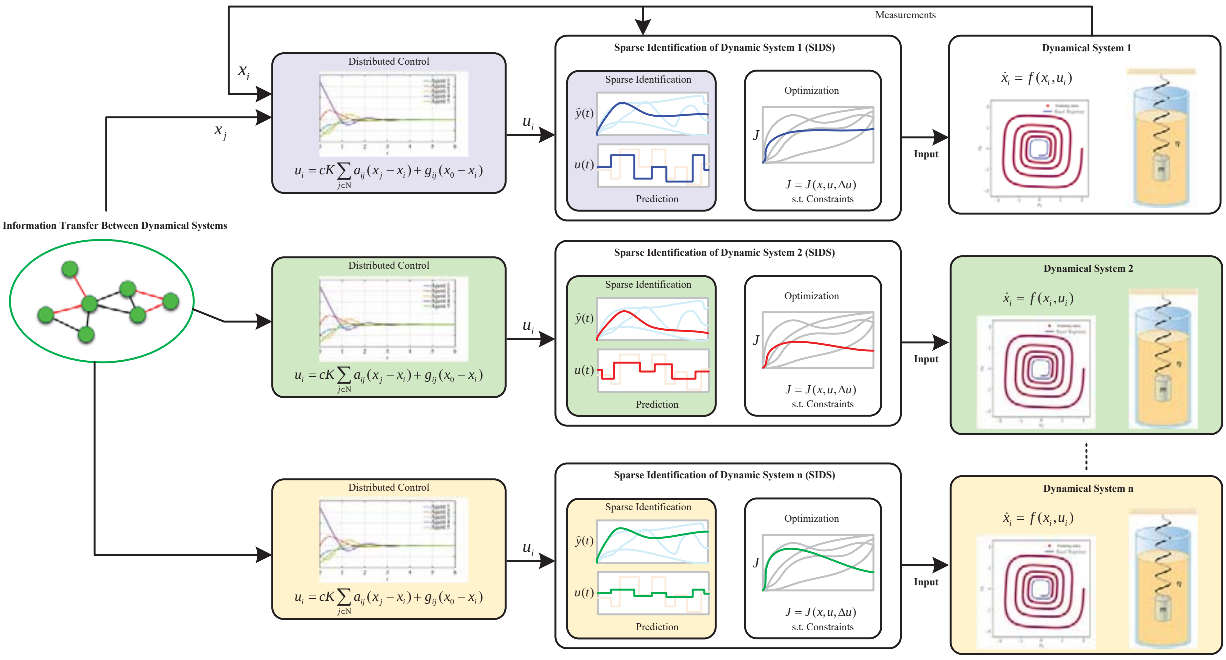

Fig. 2 Structure of the proposed model-free distributed control.

$$\frac{d}{dt}\dot{\mathbf{x}} = A_i\mathbf{x} + B_i\mathbf{u} = \begin{bmatrix} -0.1 & 2 \\ -2 & -0.1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} \mathbf{u} \quad (13)$$

III. Distributed Control Design

In this section, the distributed control design for multiple dynamical systems with unknown dynamics is explored. This is a practical problem in real-world applications and often the exact dynamics of the dynamical system of interest are not exactly known. Classical distributed control design requires the detailed dynamics of each distributed unit in order to guarantee the optimality and stability of the design. It is interesting to investigate whether the learned dynamics using sparse identification can directly be used to design distributed controllers with good performance. The predicted dynamics of the controlled undamped oscillator are used to design a distributed controller that can synchronize the response of controllable variables (i.e., x_1 in this example) using a consensus following protocol and track a global reference using a consensus tracking protocol. The structure of the proposed model-free distributed control design is depicted in Fig. 2. Assuming there exits n damped oscillators with different dynamics (heterogeneous), a sparse identification engine can be dedicated to each dynamical system to identify its dynamics using available measurements. Once the dynamics are identified, a control input can be designed for the predicted system using communications between the dynamical systems (information sharing following a communication graph) and the designed input is supplemented to the actual system with unknown dynamics.

A. Graph Theory

To design consensus algorithms for damped oscillators, multi-agent system (MAS) theory is implemented by considering each damped oscillator as an agent. Let us assume an undirected graph $\mathcal{G}$ with its vertex set $\mathcal{V}$ and edge set $\mathcal{E}$, for the communication system between dynamical systems. In this notation, a vertex represents an agent and an edge $(k, j) \in \mathcal{E}$ corresponds to the connection between agents k and j. The neighboring set of agent k is denoted by $\mathcal{N}_k \triangleq \{j \in \mathcal{V} : (k, j) \in \mathcal{E}\}$. Furthermore, let a_{kj} denote the kj^{th} element of the adjacency matrix $\mathcal{A}$ of $\mathcal{G}$, i.e. $a_{kj} = 1$ if $(k, j) \in \mathcal{E}$ and $a_{kj} = 0$ if $(k, j) \notin \mathcal{E}$. Then the degree matrix of $\mathcal{G}$ is denoted by $\mathcal{D} = \text{diag}\{d_k\}_{k=1,\ldots,n}$, where $d_k \triangleq \sum_{j\in\mathcal{N}_k} a_{kj}$. Consequently, the Laplacian matrix $\mathcal{L}$ associated to $\mathcal{G}$ is defined by $\mathcal{L} = \mathcal{D} - \mathcal{A}$.

B. Control Design

Using the relative state information $x_i \in \mathbb{R}^n$, the control input $u_i \in \mathbb{R}^k$ for the average consensus protocol is defined as:

$$u_i = K \sum_{j\in\mathcal{N}_i} a_{ij}(x_j - x_i) \quad (14)$$

where $K \in \mathbb{R}^{k\times n}$ is a control gain that can be designed in the sense that all the states in the subsystem converge to the same value asymptotically when:

$$K = B^T P \quad (15)$$

where B is the input matrix and P is a positive definite matrix that satisfies:

$$A^T P + PA - 2\beta PBB^T P + \xi PP + \frac{\eta_0^2}{\xi} I_n < 0 \quad (16)$$

where $\eta = [\eta_1^T, \eta_2^T, \ldots, \eta_N^T]$ and $\eta_i = x_i - \sum_{j=1}^{N} r_j x_j$ (r_j is the left eigenvector of graph laplacian matrix) is the state disagreement vector, ξ is any positive real number, and $\beta = \min\{\lambda_1, \ldots, \lambda_{n_\lambda}, \alpha_1, \ldots, \alpha_{n_\mu}\}$ (λ_i are n_λ real eigenvalues of Laplacian matrix, and $\alpha_i \pm \beta_i$ are n_μ complex eigenvalues of Laplacian matrix. In the above equation, η_0 is defined as:

$$\eta_0 = 2N\text{sing}(T)\text{sing}(T^{-1})\left\|r\right\|_2 \tag{17}$$

where $N = 1 + n_\lambda + 2n_\mu$ and $\text{sing}(T)$ is the largest singular value of a nonsingular matrix T satisfying $T^{-1}LT = J$, where J includes a diagonal matrix with 0 as its first element and other diagonal terms include the eigenvalues of Laplacian matrix as in (18):

$$J = \begin{bmatrix} 0 & \ldots & \ldots & \ldots & \ldots & \ldots & 0 \\ 0 & \lambda_1 & \ldots & \ldots & \ldots & \ldots & 0 \\ 0 & \ldots & \ddots & \ldots & \ldots & \ldots & 0 \\ 0 & \ldots & \ldots & \lambda_{n_\lambda} & \ldots & \ldots & 0 \\ 0 & \ldots & \ldots & \ldots & \lambda_1 & \ldots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \ldots & \ldots & \ldots & \ldots & \ldots & \lambda_{n_\mu} \end{bmatrix} \tag{18}$$

The proof for the above conditions can be found in [14]. If the objective is to design a consensus tracking problem so that state x_i of the dynamical system tracks a reference x_0, the input can be designed as:

$$u_i = K\sum_{j\in\mathcal{N}_i} a_{aj}(x_j - x_i) - K_0 a_{0i}(x_0 - x_i) \tag{19}$$

where a_{0i} denotes the adjacency element between the leader (one of the agents that receive reference information and shares with its neighbors) and other agents (followers), which is 1 if there is a connection between agent i and the leader (agent 0), and is 0 otherwise. The gain K_0 can be designed similar to gain K. The above input supplemented to the dynamical system guarantees the error between state x_i and reference x_0 will approach zero, i.e., $x_i \longrightarrow x_0$.

IV. Case Studies

To validate the effectiveness of the proposed model-free distributed control, three controlled damped oscillators have been considered. The actual but assumed unknown dynamics of the heterogeneous oscillators are governed by their state matrices in the following:

$$A_1 = \begin{bmatrix} -0.1 & 2 \\ -2 & -0.1 \end{bmatrix},\ B_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \tag{20}$$

$$A_2 = \begin{bmatrix} -0.2 & 3 \\ -2 & -0.41 \end{bmatrix},\ B_2 = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \tag{21}$$

$$A_3 = \begin{bmatrix} -0.15 & 1 \\ -1 & -0.21 \end{bmatrix},\ B_3 = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \tag{22}$$

First, each heterogeneous oscillator is supplemented with a sparse identification model that received its measurements and identifies the governing dynamics. For $\Theta(\mathbf{X},\mathbf{U})$, polynomials up to degree 3 are considered as in (23). The resulting $\mathbf{\Gamma_i}$ matrices for these heterogeneous oscillators are listed in Table I. Once the dynamics of the oscillators are identified, the distributed controllers are designed in two case studies using the ideas in Section II.

TABLE I
PERFORMANCE MEASURES OF THE PROPOSED OPTIMIZATION MODEL

Row	Γ_1	$\mathbf{\Gamma_2}$	$\mathbf{\Gamma_3}$
Row 1	[0 0]	[0 0]	[0 0]
Row 2	[-0.149 1.997]	[-0.32 2.99]	[-0.147 0.997]
Row 3	[-1.939 -0.11]	[-1.88 -0.412]	[-1.012 -0.176]
Row 4	[1.043 0]	[1.029 0]	[0.995 0]
Rows 5-13	[0 0]	[0 0]	[0 0]

Fig. 3 depicts the simulation results for distributed control of heterogeneous damped oscillators, where Fig. 3 (a) depicts the control design performance (convergence to desired values) on the actual systems (physical models) and Fig. 3 (b) illustrates the control design on the predicted sparse identification dynamics. The first subplots on the top depict the dynamics of state 1 (x_1) for these three oscillators when no controller is supplemented to the system. Due to the heterogeneity of the dynamical systems, the responses of these systems show different settling time and overshoots (due to various initial conditions), but they all stabilize at their equilibrium point (0) eventually. It can also be confirmed that the learned dynamics exactly match with the physical dynamics of the systems, denoting a successful identification of system dynamics. Subplots in the middle demonstrate the effectiveness of consensus following protocol in synchronizing these damped oscillators. Due to the implementation of the consensus protocol in (14), all damped oscillators reach to their equilibrium at the same settling time and frequencies. It can also be confirmed that the learned dynamics (left subplot) exactly matches the distributed control design on the physical dynamics. Finally, the consensus tracking protocol was supplemented to the system by enabling oscillator 1 to be leader with a setpoint of $x_0 = 1$. As a result of the consensus tracking protocol in (19), all the units will track the same reference and the equilibrium point of all damped oscillators changes to 1 (reference) confirming a successful tracking control design. It is noted that the consensus design on learned dynamics (right subplots) perfectly matches with the design on the physical system without knowing the dynamics of the physical system. The results suggest a potential for implementation of distributed controllers for more complex cyber-physical systems, i.e. robots, air crafts, and buildings, without complex modeling procedures.

V. Conclusion

In this paper, a model-free distributed control design of dynamical systems was studied. Using sparse identification of system dynamics with control along with available

$$\Theta(\mathbf{X},\mathbf{U}) = \begin{bmatrix} 1 & \mathbf{x_1} & \mathbf{x_2} & \mathbf{u} & \mathbf{x_1}^2 & \mathbf{x_1x_2} & \mathbf{x_2}^2 & \mathbf{x_1}^3 & \mathbf{x_1}^2\mathbf{x_2} & \mathbf{x_1x_2}^2 & \mathbf{x_2}^3 & \mathbf{x_1u} & \mathbf{x_2u} \end{bmatrix} \tag{23}$$

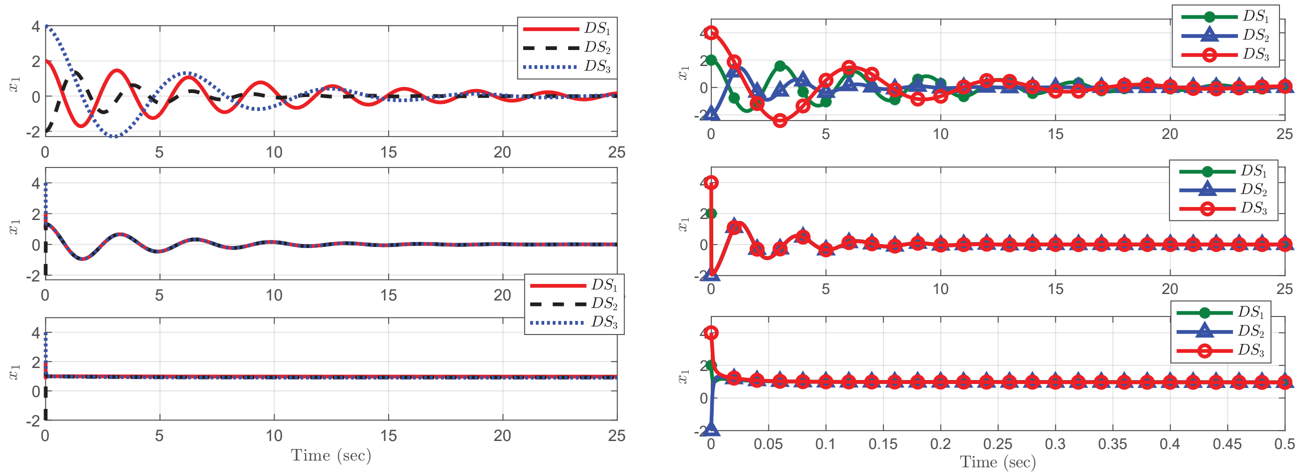

Fig. 3 Comparison between the distributed control design on the actual dynamics of damped oscillators (a) and predicted dynamics using sparse identification (b)

measurements, dynamics of the system were predicted with candidate polynomial functions. The learned dynamics were then used to design a distributed controller for consensus tracking and following problems. The proposed research demonstrates the effectiveness of the sparse identification technique for distributed control design of linear and nonlinear systems. Such formulation can significantly enhance the control design issue of complex dynamical systems. Future research will focus on the application of distributed consensus design using sparse identification technique in smart grid applications.

REFERENCES

[1] M. Liu, L. Tan, and S. Cao, "Method of dynamic mode decomposition and reconstruction with application to a three-stage multiphase pump," *Energy*, vol. 208, p. 118343, 2020.

[2] H. Lu and D. M. Tartakovsky, "Prediction accuracy of dynamic mode decomposition," *SIAM Journal on Scientific Computing*, vol. 42, no. 3, pp. A1639–A1662, 2020.

[3] C. Folkestad, D. Pastor, I. Mezic, R. Mohr, M. Fonoberova, and J. Burdick, "Extended dynamic mode decomposition with learned koopman eigenfunctions for prediction and control," in *2020 american control conference (acc)*. IEEE, 2020, pp. 3906–3913.

[4] M. Al-Gabalawy, "Deep learning for koopman operator optimal control," *ISA transactions*, 2021.

[5] A. Mauroy, I. Mezić, and Y. Susuki, *The Koopman Operator in Systems and Control: Concepts, Methodologies, and Applications*. Springer Nature, 2020, vol. 484.

[6] M. Schmidt and H. Lipson, "Distilling free-form natural laws from experimental data," *science*, vol. 324, no. 5923, pp. 81–85, 2009.

[7] S. L. Brunton, J. L. Proctor, and J. N. Kutz, "Discovering governing equations from data by sparse identification of nonlinear dynamical systems," *Proceedings of the national academy of sciences*, vol. 113, no. 15, pp. 3932–3937, 2016.

[8] ——, "Sparse identification of nonlinear dynamics with control (sindyc)," *IFAC-PapersOnLine*, vol. 49, no. 18, pp. 710–715, 2016.

[9] E. Kaiser, J. N. Kutz, and S. L. Brunton, "Sparse identification of nonlinear dynamics for model predictive control in the low-data limit," *Proceedings of the Royal Society A*, vol. 474, no. 2219, p. 20180335, 2018.

[10] D. Ding, Q.-L. Han, Z. Wang, and X. Ge, "A survey on model-based distributed control and filtering for industrial cyber-physical systems," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 5, pp. 2483–2499, 2019.

[11] A. Savitzky and M. J. Golay, "Smoothing and differentiation of data by simplified least squares procedures." *Analytical chemistry*, vol. 36, no. 8, pp. 1627–1639, 1964.

[12] R. W. Schafer, "What is a savitzky-golay filter? [lecture notes]," *IEEE Signal Processing Magazine*, vol. 28, no. 4, pp. 111–117, 2011.

[13] S. Larsson and V. Thomée, *Partial differential equations with numerical methods*. Springer, 2003, vol. 45.

[14] Z. Ding, "Consensus control of a class of nonlinear systems," in *2013 9th Asian Control Conference (ASCC)*. IEEE, 2013, pp. 1–6.