Data-Driven Classification, Reduction, Parameter Identification and State Extension in Hybrid Power Systems

Andrija T. Sarić, *Member, IEEE*, Mark K. Transtrum, and Aleksandar M. Stanković, *Fellow, IEEE*

***Abstract*—The paper describes a manifold learning-based algorithm for big data classification and reduction, as well as parameter identification in real-time operation of a power system. Both black-box and gray-box settings for SCADA- and PMU-based measurements are examined. Data classification is based on diffusion maps, where an improved data-informed metric construction for partition trees is used. Data classification and reduction is demonstrated on the measurement tensor example of calculated transient dynamics between two SCADA refreshing scans. Interpolation/extension schemes for state extension of restriction (from data to reduced space) and lifting (from reduced to data space) operators are proposed. The method is illustrated on the single-phase Motor D example from a very detailed WECC load model, connected to the single bus of a real-world 441-bus power system.**

***Index Terms*—Data classification, Dynamic model reduction, Parameter identification, State extension, Manifold learning, Diffusion map, WECC model.**

I. Introduction

EFFICIENT processing of massive high-dimensional data sets and the extraction of actionable information are contemporary challenges in modern power systems. Many classical data processing algorithms have a computational complexity that grows exponentially with the size of data (the so-called "curse of dimensionality") [1]. A growing number of references have proposed different methods for quick extraction of useful information from large datasets to improve the reliability, efficiency, and flexibility of the grid [2]–[6]. The one aspect that is often missing stems from the fact that many variables contained in each data point are correlated (locally or globally), suggesting that a possibly large data set has a much lower intrinsic dimensionality. It should thus be possible to obtain a low-dimensional representation of observation (measurement) samples. When correlations between variables are only local, classical (and global) dimension reduction methods like principal component analysis and multidimensional scaling typically do not achieve an efficient dimension reduction.

Diffusion maps applied in the context of manifold learning are becoming an increasingly popular means to overcome such problems [7]. They implement a promising form of nonlinear dimensionality reduction but may prove sensitive to the way the data points were sampled. More precisely, if the data are assumed to approximately lie on a manifold, then eigenmap representation depends on the density of the points on this manifold [8]. This is important in cases when it is needed to merge data produced by the same source but acquired with different sensors ("data matching"), or where different sampling rates are present (both instances are typical in power systems). In these cases, it is necessary to have a canonical representation of the data that retains the intrinsic constraints of the samples (for example, provided by a manifold geometry). Data matching establishes a correspondence between two sets obtained from the same source. This aim is achieved by the creation of so-called partition trees along different axes.

Initial results with data classification via diffusion maps applied to electromagnetic transient analysis in power systems are presented in [9].

We consider a very general problem formulation here—multiple measurements at different parameter settings, denoted as "trials" from an unknown, nonlinear, parametrically dependent dynamical system. The problem setting includes a large ensemble of short time series indexed by the label of the trial as well as by the label of the measurement channel, while each time series is parametrized by time. However, the knowledge of how many and what parameters the system has and the actual settings at which the trials are performed is **hidden**. Furthermore, we do not know how many and which state variables the system has, or what functions of the state variables we measure. We only know what each channel recorded, at each trial, as a function of time (for a short time). Using the similarity between individual pairs of this large ensemble of short time series as our only tool, we identify a set of relevant parameters and a set of relevant state variables. Later we demonstrate on examples that the

Manuscript received April 9, 2020; revised July 15, 2020; accepted September 13, 2020. Date of publication September 28, 2020; date of current version April 19, 2021. This work was supported in part by ARPA_E under contract DE-AR0000223, in part by CURENT Engineering Research Center of the National Science Foundation, and in part by the Department of Energy under NSF Award Number EEC-1041877. Paper no. TPWRS-00588-2020. *(Corresponding author: Aleksandar M. Stanković.)*

Andrija T. Sarić is with the Department for Power, Electronic and Communication Engineering of the Faculty of Technical Sciences, University of Novi Sad, Novi Sad 21000, Serbia (e-mail: asaric@uns.ac.rs).

Mark K. Transtrum is with the Department of Physics and Astronomy, Brigham Young University, Provo, UT 84602 USA (e-mail: mktranstrum@byu.edu).

Aleksandar M. Stanković is with the Department of Electrical Engineering and Computer Science of Tufts University, Medford, MA 02155 USA (e-mail: astankov@ece.tufts.edu).

Color versions of one or more of the figures in this article are available online at https://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TPWRS.2020.3027249

methodology we explore also can deal with bad data. One of our larger goals is to integrate the proposed algorithms with traditional, physics-derived models, as shown in [10].

This paper focuses on both black-box and gray-box settings, where both supervisory control and data acquisition (SCADA)-based and phasor measurement unit (PMU)-based measurements are available. Traditional SCADA-based measurements, obtained from asynchronous remote terminal units (RTUs), are typically collected in time steps of over 2–10 s. With the availability of PMUs, the synchronized measurements can be taken at rates of 30–120 samples/s. Such measurements are typically accompanied by timestamps, thus giving insight into actual dynamics of the power system.

The proposed data classification is demonstrated on a very detailed Western Electricity Coordinating Council (WECC) load model example, where an extension to simpler representations of dynamical components (for example, synchronous generators) is straightforward. In this paper, typical real-time disturbances, like daily load/generation variations are considered. Such conditions are very challenging from the classification standpoint. In cases of major disturbances (like three-phase short circuits, branch outages, etc.) it is obvious in practice that the responses will be classified into the different pattern(s).

The paper is organized as follows: Section II describes the power system dynamic model (including typical cases for measurement types and model availability) and the measurement tensor; in Section III details are presented about data classification by the manifold learning algorithm (diffusion maps and affinity matrix, as well as the informed metric); parameter identification by nonlinear constrained optimization is proposed in Section IV, while the transient analysis for gray-box modeling is given in Section V; interpolation/extension schemes for input measurements-output eigenvectors (the Nyström extension [11], the radial basis function and restriction/lifting operators [12], and a Kriging algorithm [12]) are described in Section VI; the proposed method is applied to the single-phase *Motor D* from detailed WECC dynamic load model in the 441-bus real-world test system in Section VII; finally Section VIII presents conclusions, while the *Appendix* provides some details about the WECC load model.

II. Model Description

A. Dynamic Model

A data-driven framework for classification of time-dependent measurements (observations) in dynamical power system is built upon nonlinear differential and algebraic equations (DAEs), respectively

$$\mathrm{d}\boldsymbol{x}/\mathrm{d}t = \boldsymbol{f}\left(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{p}, t\right) \tag{1}$$

$$0 = \boldsymbol{g}\left(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{p}, t\right) \tag{2}$$

where $\boldsymbol{x}$ is the vector of state variables, $\boldsymbol{z}$ are the algebraic variables, $\boldsymbol{p}$ are parameters, and t is the (scalar) time variable.

System measurements are assumed to be of the form

$$\boldsymbol{y} = \boldsymbol{h}\left(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{p}, t\right) \tag{3}$$

We assume that the power system's evolution ($\boldsymbol{f}$ and $\boldsymbol{g}$) and measurement functions ($\boldsymbol{h}$), as well as state ($\boldsymbol{x}$) and algebraic ($\boldsymbol{z}$) variables, are unknown (for the black-box settings), or partially known (for the gray-box settings). We have only measurements ($\boldsymbol{y}$) labeled by time (t) and initial values of parameters ($\boldsymbol{p}$). We want to characterize the power system dynamics by systematically organizing the observations (collected over periodic measurements) of its outputs ($\boldsymbol{y}$).

B. Black-Box and Gray-Box Models

Depending on the level of power system equipment with measurement sensors (SCADA and/or PMUs) at the *node where the WECC dynamic model is connected and in the rest of the power system*, as well as the availability of the power system model (black-box or gray-box models), the following typical cases may be identified:

Case 1. SCADA Measurements, the Black-Box Model: Active/reactive (P/Q) load and voltage magnitude (V) measurements are available. The availability of measurements from the rest of the power system is irrelevant since there is no available information about the full power system model (the black-box model). The time frame for data classification may be, for example, one hour to detect slow variations of load patterns. This case is studied extensively in [13].

Case 2. SCADA Measurements, the Gray-Box Model: The P/Q and V measurements at the point where the WECC dynamic load is connected and measurements in the rest of the power system (depending on the measurements redundancy) are available. A complete power system dynamic model with uncertain parameters is also available. In this case, the static state estimation is performed in observable areas (as a pre-filtering step). As a result, the P/Q and V/θ (θ is voltage angle) filtered estimates may be used for parameter identification (by a nonlinear constrained optimization). The time frame for data classification may be as in *Case 1*. Alternatively, the time frame for data classification may be shortened to several seconds (or the time between two consecutive SCADA snapshots) to detect fast variations of input patterns, where the unavailable dynamics can be obtained from the transient analysis (TA) tool—see Section V for details.

Case 3. PMU Measurements, the Black-Box Model: The P/Q and V/θ measurements are available. In this setup, the availability of measurements from the rest of the power system is again irrelevant, since there is no available information about the full power system dynamic model. Then, the available measurements may be used directly for the parameter identification (by a nonlinear constrained optimization from Section IV). The time frame for data classification may be, for example, several seconds to detect fast variations of load patterns.

Case 4. PMU Measurements, the Gray-Box Model: The P/Q and V/θ measurements and complete power system dynamic model (with uncertain parameters) are available. Then, the dynamic state estimation (like Kalman filtering) is performed (as a pre-filtering step). As a result, the P/Q and V/θ filtered measurements may be used for the parameter identification (by the nonlinear constrained optimization from Section IV). Optionally, TA may be performed to provide detailed transients

for all the state/algebraic variables (for different purposes). The time frame for data classification may be as in *Case 3*.

Case 5. Mixed SCADA/PMU Measurements, the Gray-Box Model: Different arrangements of this case may occur. For example, in power systems with mixed SCADA/PMU measurements, as well as the gray-box model (mixed *Case 2* and *Case 4*), the time frame for data classification can be the time between two consecutive SCADA scans (2−10 s).

C. Measurement Tensor

We define the measurement tensor ($\boldsymbol{Y}$) as follows.

Formally, let $\mathcal{M}$ denote an ensemble of $N_{\boldsymbol{m}}$ sets of observations (measurements) and $\mathcal{W}$ denote an ensemble of $N_{\boldsymbol{w}}$ sets of time windows for observations.

For each $\boldsymbol{m} \in \mathcal{M}$ and $\boldsymbol{w} \in \mathcal{W}$ we observe a trajectory $\boldsymbol{y}(\boldsymbol{m}, \boldsymbol{w}, t)$ of length N_t of the system variables, where $t = 1,\ 2,\ \ldots, N_t$ denotes the time samples.

Let $\boldsymbol{Y}$ denote the entire 3-D tensor of observations (measurements) $\boldsymbol{y}(\boldsymbol{m}, \boldsymbol{w}, t)$, with dimensions $N_{\boldsymbol{m}} \times N_{\boldsymbol{w}} \times N_t$, where $N_{\boldsymbol{m}}$ denotes the number of measurements and $N_{\boldsymbol{w}}$ denotes the number of observation time windows.

III. Data Classification by Manifold Learning Algorithm

We define the trajectory of measurements in the time windows axis as [14], [15]

$$\boldsymbol{y}_{\boldsymbol{w}} = \{\boldsymbol{Y}(\boldsymbol{m}, \boldsymbol{w}, t) | \forall_{\boldsymbol{w}}, \forall_t\}, \boldsymbol{w} \in \mathcal{W} \tag{4}$$

for each of N_w vectors.

Similarly, let $\boldsymbol{y}_{\boldsymbol{m}}$ and $\boldsymbol{y}_t$ be the samples as seen from the standpoints of the measurement and time axes, respectively, defined by

$$\boldsymbol{y}_{\boldsymbol{m}} = \{\boldsymbol{Y}(\boldsymbol{m}, \boldsymbol{w}, t) | \forall \boldsymbol{m}, \forall t\}, \boldsymbol{m} \in \mathcal{M} \tag{5}$$

$$\boldsymbol{y}_t = \{\boldsymbol{Y}(\boldsymbol{m}, \boldsymbol{w}, t) \forall \boldsymbol{m}, \forall \boldsymbol{w}\}, t = 1,\ 2,\ \ldots, N_t \tag{6}$$

All data are processed three consecutive times, each time from a different viewpoint (measurements axis, time windows axis, and time axis, respectively).

A. Diffusion Maps and Affinity Matrix

Data-driven parameterization is based on a kernel. From the trials axis [$\boldsymbol{y}$ in (4)-(6)] point of view, a typical kernel is defined by

$$\kappa\ (\boldsymbol{y}_i, \boldsymbol{y}_j) = e^{-\left(\frac{\|\boldsymbol{y}_i - \boldsymbol{y}_j\|}{\varepsilon}\right)^2},\ \forall \boldsymbol{y}_i, \boldsymbol{y}_j \in \boldsymbol{Y} \tag{7}$$

where the Gaussian function induces a sense of locality relative to the kernel scale ε [14, eq. (4)].

To aggregate the pairwise affinities comprising the kernel into a global parameterization, the eigenvalue decomposition is applied to the kernel, and the eigenvalues and eigenvectors are used to construct the desired parameterization by diffusion maps [14].

One class of data-driven methods for analyzing complex datasets is manifold learning based on the diffusion maps [8]. The main assumption is that data are constrained to lie on or around a low-dimensional manifold. A common practice is to interpret the set of samples and the affinity matrix ($\boldsymbol{W}$) as a graph, where the samples are the graph nodes and the affinity matrix determines the weights of the edges.

Given set ($\boldsymbol{y}$) of N samples of observations ($N_{\boldsymbol{m}}$, $N_{\boldsymbol{w}}$, or N_t in Section II-C), let $\boldsymbol{W}$ is $(N \times N)$ pairwise affinity matrix, with (i, j)-th entry (for $\boldsymbol{y}_i$ and $\boldsymbol{y}_j$ samples) defined by

$$W_{\boldsymbol{ij}} = e^{-\left(\frac{\|\boldsymbol{y}_i - \boldsymbol{y}_j\|}{\varepsilon}\right)^2} \tag{8}$$

A weight of 1 indicates that two samples ($\boldsymbol{y}_i$ and $\boldsymbol{y}_j$) are identical, while weights close to 0 indicates that two samples are very dissimilar. In our case, a node (folder) $\boldsymbol{y}_i$ is connected to another node (folder) $\boldsymbol{y}_j$ by an edge with weight $W_{\boldsymbol{ij}}$ (see Section III-B).

The next step aggregates the pairwise affinities/graph connections into a global parameterization. The affinity matrix ($\boldsymbol{W}$) is normalized to a row stochastic matrix $\boldsymbol{A} = \boldsymbol{D}^{-1}\ \boldsymbol{W}$, where $\boldsymbol{D}$ is a diagonal matrix with elements equal to the sum of rows of $\boldsymbol{W}$. In the graph interpretation, $\boldsymbol{A}$ can be seen as a transition probability matrix defining a Markov chain on the graph, where A_{ij} is the probability to "jump" from the node (folder) $\boldsymbol{y}_i$ to the node (folder) $\boldsymbol{y}_j$ in one Markov chain step [14].

The diffusion map is used to calculate the informed metric (10) inside the partition tree algorithm (see Section III-B).

To aggregate the pairwise affinities comprising the kernel into a global parameterization, the eigenvalue decomposition is applied to the kernel (row stochastic matrix, $\boldsymbol{A}$), and the eigenvalues and eigenvectors are used to construct the desired parameterization [12], [14], [15]. Let λ_k denote the eigenvalues ordered in decreased order, and let $\boldsymbol{\psi}_k$ [$\psi_k(t)$] denote the corresponding (right) eigenvector. Matrix $\boldsymbol{A}$ is row stochastic, and its largest eigenvalue is $\lambda_0 = 1$, corresponding to the trivial eigenvector $\boldsymbol{\psi}_0$ with all-ones. Since, λ_0 and $\boldsymbol{\psi}_0$ do not carry relevant information on the data, they are ignored. The diffusion map embedding of the samples is defined as the following nonlinear map [14, eq. (9)]

$$\{\boldsymbol{\lambda}, \boldsymbol{\psi}\} \rightarrow (\lambda_1 \boldsymbol{\psi}_1, \ldots, \lambda_k \boldsymbol{\psi}_k, \ldots, \lambda_K \boldsymbol{\psi}_K) \tag{9}$$

where K denotes by the user-selected number of largest eigenvalues.

Note that the Euclidean distance between the embedded samples approximates the diffusion distance (a distance defined by the induced transition probabilities) is closely related to the geodesic distance on the assumed underlying manifold. The eigenvectors $\boldsymbol{\psi}_k$ form an orthonormal basis for any real function defined on the sample set $\{\boldsymbol{y}\}$.

From three separate diffusion maps for data sets $\boldsymbol{y}_{\boldsymbol{m}}$, $\boldsymbol{y}_{\boldsymbol{w}}$, and $\boldsymbol{y}_t$ we may obtain three mappings as in (9), denoting the associate eigenvectors by $\{\boldsymbol{\psi}_k^{\mathcal{M}}\}$, $\{\boldsymbol{\psi}_k^{\mathcal{W}}\}$, and $\{\boldsymbol{\psi}_k^{t\ =\ 1,\ 2,\ \ldots,N_t}\}$; $k = 1,\ 2,\ \ldots, K$, respectively. However, such mappings do not take into account the strong correlation and dependencies among the different time windows (for example, dynamics of the power system between two measurement snapshots), or the parameter variations with the dynamics of state and algebraic variables, which is typical in dynamical systems (such as a power system).

To incorporate such dependencies, an *informed (distance) metric* between samples in the different axes may be derived from observations.

B. Informed Metric

The construction of the informed metric is described on the time window axis ($\boldsymbol{w} \in \mathcal{W}$) example; extensions to other axes are straightforward. The essence is the definition of a meaningful notion of distance between samples. The informed metric $\|\boldsymbol{y}_{\boldsymbol{w}i} - \boldsymbol{y}_{\boldsymbol{w}j}\|^{\mathcal{W}}$, where the superscript $\mathcal{W}$ indicates that it is an informed norm between the samples—informed from the time window viewpoint in this case. The construction of the metric is iterative. In each iteration, the dependencies between the samples are gradually revealed from observations and, in turn, used to build a refined informed metric (for details of the construction procedure see in [14, Appendix 2]). In the first iteration, the construction of the informed metric $\|\boldsymbol{y}_{\boldsymbol{w}i} - \boldsymbol{y}_{\boldsymbol{w}j}\|^{\mathcal{W}}$ uses as an initial input two noninformed metrics, defined on the time windows axis ($\boldsymbol{w} \in \mathcal{W}$) and on the time axis ($t = 1, 2, \ldots, N_t$). Possible choices for such metrics are the Euclidean metric or a metric derived from the cosine between vectors [14]. The construction is implemented by decomposing the metric into the following general form

$$\|\boldsymbol{y}_{\boldsymbol{w}i} - \boldsymbol{y}_{\boldsymbol{w}j}\|^{\mathcal{W}} = \|\boldsymbol{y}_{\boldsymbol{w}i} - \boldsymbol{y}_{\boldsymbol{w}j}\|_1 + \gamma\|\mathcal{F}^{\mathcal{W}}(\boldsymbol{y}_{\boldsymbol{w}i}) - \mathcal{F}^{\mathcal{W}}(\boldsymbol{y}_{\boldsymbol{w}j})\|_1 \quad (10)$$

where $\|\cdot\|_1$ is the ℓ_1 norm, $\gamma > 0$ is a weighting factor, and $\mathcal{F}^{\mathcal{W}} : \mathbb{R}^{N_{\boldsymbol{w}} \times N_t} \to \mathbb{R}^D$ is a feature function (i.e., a measure of the transformation from data to reduced space—*D*).

The 2-D (classical) transform is implemented using a set of basis functions $(g_{\ell,\ell'})$ and the transform is given by a collection of the linear projections of the data on that set of basis functions

$$\mathcal{F}^{\mathcal{W}}(\boldsymbol{y}_{\boldsymbol{w}}) = \{g_{\ell,\ell'}, \boldsymbol{y}_{\boldsymbol{w}} \forall \ell\} \quad (11)$$

where for the time windows axis, the basis functions are defined on $\mathcal{W} \times \{1, 2, \ldots, N_t\}$ and the inner product is

$$\langle g_{\ell,\ell'}, \boldsymbol{y}_{\boldsymbol{w}} \rangle = \sum_{\boldsymbol{w}\mathcal{W}} \sum_{t=1,\ 2,\ \ldots,N_t} g_{\ell,\ell'}(\boldsymbol{w}, t)\, \boldsymbol{y}_{\boldsymbol{w}}(\boldsymbol{m}, \boldsymbol{w}, t) \quad (12)$$

However, the classical transforms above (such as, for example, Fourier transform, Wavelet transform etc.) are linear and local, and their basis functions $g_{\ell,\ell'}$ are fixed and not data-adaptive. This paper applies a transform based on data-driven partition trees [16]–[18]. By using the initial non-informed metrics, a multi-level clustering approach employing a hierarchical partition tree of both the measurements and time windows is applied. Each partition tree is composed of $L + 1$ levels, where a partition of samples is defined for each level $0 \le l \le L$. The partition at the l-th level consists of $n(l)$ mutually disjoint nonempty subsets of samples—folders. At the bottom level of the tree ($l = 0$), each sample belongs to its own tree leaf (finest partition). At each level, moving up, folders are merged to create larger and larger folders, until reaching the root of the tree, where all the samples belong to a single root folder ($l = L$). These trees are used to define an over-complete set of basis functions (defined for each folder I_ℓ in the time windows axis tree and for each folder $J_{\ell'}$ in the time axis tree) as the indicator function for the samples in these folders as

$$g_{\ell,\ell'}(\boldsymbol{w}, t) = \begin{cases} 1 & \boldsymbol{w} \in I_\ell,\ t \in J_{\ell'} \\ 0 & \text{otherwise} \end{cases} \quad (13)$$

For the basis functions above, a transform based on the earth mover's distance is formulated. For details of the construction procedure see in [14, Appendix 2].

IV. Parameter Identification

Parameter identification is described for *Case 4*: PMU, the gray-box model (Section II-B).

Parameter fitting to the observations is based on the modified nonlinear optimization from [19]

$$\hat{\boldsymbol{p}} = \min \left\{ \frac{1}{2} \sum_{\boldsymbol{m}=1}^{N'_{\boldsymbol{m}}} W_{\boldsymbol{m}} \sum_{\boldsymbol{w}=1}^{N'_{\boldsymbol{w}}} \sum_{t=1}^{N_t} r^2_{\boldsymbol{m}\boldsymbol{w}t}(\boldsymbol{p}) \right\} \quad (14)$$

subject to the parameter's lower/upper bounds

$$\boldsymbol{p}^{min} \le \boldsymbol{p} \le \boldsymbol{p}^{\max} \quad (15)$$

where:

$N'_{\boldsymbol{m}}$	− number of analyzed measurements ($N'_{\boldsymbol{m}} \le N_{\boldsymbol{m}}$);
$N'_{\boldsymbol{w}}$	− number of analyzed time windows patterns ($N'_{\boldsymbol{w}} \le N_{\boldsymbol{w}}$);
N_t	− number of elements in time axis of measurement tensor;
$W_{\boldsymbol{m}}$	− elements of $(N'_{\boldsymbol{m}} \cdot N'_{\boldsymbol{w}} \cdot N_t)$-dimensional diagonal weighting matrix for measurement set ($\boldsymbol{m} \in \mathcal{M}$);
$r_{\boldsymbol{m}\boldsymbol{w}t}(\boldsymbol{p}) = h_{\boldsymbol{m}\boldsymbol{w}t} - h_{\boldsymbol{m}\boldsymbol{w}t}(\boldsymbol{p})$	is the error (residual) for the $\boldsymbol{m}$-th measurement function, $\boldsymbol{w}$-th time windows pattern, and t-th time point;
$h_{\boldsymbol{m}\boldsymbol{w}t},\ h_{\boldsymbol{m},\boldsymbol{w},t}(\boldsymbol{p})$	− measurement functions for the basic and perturbed parameters, respectively.

Introducing cumulative vectors we have:

- for N_t analyzed time points:

$$\boldsymbol{r}_{\boldsymbol{m}\boldsymbol{w}}(\boldsymbol{p}) = \begin{bmatrix} r_{\boldsymbol{m}\boldsymbol{w}1}(\boldsymbol{p}) \\ \vdots \\ r_{\boldsymbol{m}\boldsymbol{w}t}(\boldsymbol{p}) \\ \vdots \\ r_{\boldsymbol{m}\boldsymbol{w}N_t}(\boldsymbol{p}) \end{bmatrix} = \boldsymbol{h}_{\boldsymbol{m}\boldsymbol{w}} - \boldsymbol{h}_{\boldsymbol{m}\boldsymbol{w}}(\boldsymbol{p})$$

$$\boldsymbol{h}_{\boldsymbol{m}\boldsymbol{w}} = \begin{bmatrix} h_{\boldsymbol{m}\boldsymbol{w}1} \\ \vdots \\ h_{\boldsymbol{m}\boldsymbol{w}t} \\ \vdots \\ h_{\boldsymbol{m}\boldsymbol{w}N_t} \end{bmatrix}$$

$$\boldsymbol{h_{mw}}\ (\boldsymbol{p}) = \begin{bmatrix} h_{\boldsymbol{mw}1}\,(\boldsymbol{p}) \\ \vdots \\ h_{\boldsymbol{mw}t}\,(\boldsymbol{p}) \\ \vdots \\ h_{\boldsymbol{mw}N_t}\,(\boldsymbol{p}) \end{bmatrix} \boldsymbol{J^p_{mw}} = \begin{bmatrix} \boldsymbol{J^p_{mw}}1 \\ \vdots \\ \boldsymbol{J^p_{mw}}t \\ \vdots \\ \boldsymbol{J^p_{mw}}N_t \end{bmatrix}$$

- for $N'_{\boldsymbol{w}}$ time windows patterns:

$$\boldsymbol{r_m}\ (\boldsymbol{p}) = \begin{bmatrix} \boldsymbol{r_{m}}1\,(\boldsymbol{p}) \\ \vdots \\ \boldsymbol{r_{mw}}\,(\boldsymbol{p}) \\ \vdots \\ \boldsymbol{r_{m}}N'_{\boldsymbol{w}}\,(\boldsymbol{p}) \end{bmatrix} = \boldsymbol{h_m}\ - \boldsymbol{h_m}\,(\boldsymbol{p})$$

$$\boldsymbol{h_m} = \begin{bmatrix} \boldsymbol{h_{m}}1 \\ \vdots \\ \boldsymbol{h_{mw}} \\ \vdots \\ \boldsymbol{h_{m}}N'_{\boldsymbol{w}} \end{bmatrix}$$

$$\boldsymbol{h_m}\ (\boldsymbol{p}) = \begin{bmatrix} \boldsymbol{h_{m}}1\,(\boldsymbol{p}) \\ \vdots \\ \boldsymbol{h_{mw}}\,(\boldsymbol{p}) \\ \vdots \\ \boldsymbol{h_{m}}N'_{\boldsymbol{w}}(\boldsymbol{p}) \end{bmatrix} \boldsymbol{J^p_m} = \begin{bmatrix} \boldsymbol{J^p_{m}}1 \\ \vdots \\ \boldsymbol{J^p_{mw}} \\ \vdots \\ \boldsymbol{J^p_{m}}N'_{\boldsymbol{w}} \end{bmatrix},$$

- and finally for $N'_{\boldsymbol{m}}$ measurements:

$$\boldsymbol{r}\ (\boldsymbol{p}) = \begin{bmatrix} \boldsymbol{r}_1\,(\boldsymbol{p}) \\ \vdots \\ \boldsymbol{r_m}\,(\boldsymbol{p}) \\ \vdots \\ \boldsymbol{r}_{N'_{\boldsymbol{m}}}\,(\boldsymbol{p}) \end{bmatrix} = \boldsymbol{h} - \boldsymbol{h}\,(\boldsymbol{p})$$

$$\boldsymbol{h}\ = \begin{bmatrix} \boldsymbol{h}_1 \\ \vdots \\ \boldsymbol{h_m} \\ \vdots \\ \boldsymbol{h}_{N'_{\boldsymbol{m}}} \end{bmatrix}$$

$$\boldsymbol{h_m}\ (\boldsymbol{p}) = \begin{bmatrix} \boldsymbol{h}_1\,(\boldsymbol{p}) \\ \vdots \\ \boldsymbol{h_m}\,(\boldsymbol{p}) \\ \vdots \\ \boldsymbol{h}_{N'_{\boldsymbol{m}}}(\boldsymbol{p}) \end{bmatrix} \boldsymbol{J^p} = \begin{bmatrix} \boldsymbol{J^p_1} \\ \vdots \\ \boldsymbol{J^p_m} \\ \vdots \\ \boldsymbol{J^p_{N'_m}} \end{bmatrix}$$

where $\boldsymbol{J^p_{mwt}} = \frac{\partial h_{\boldsymbol{mw}t}(\boldsymbol{p})}{\partial \boldsymbol{p}}$ is a vector of sensitivities of measurement function for $\boldsymbol{m}$-th, $\boldsymbol{w}$-th, and t-th point to a set of parameters.

The necessary condition for an optimum in (14) is

$$\begin{aligned} 0 = & -\frac{\partial \boldsymbol{h}\,(\boldsymbol{p})}{\partial \boldsymbol{p}} \boldsymbol{W}\ [\boldsymbol{h} - \boldsymbol{h}\,(\boldsymbol{p})] \\ = & -\boldsymbol{J_p^{\mathrm{T}}} \boldsymbol{W}\,[\boldsymbol{h} - \boldsymbol{h}\,(\boldsymbol{p}) - \boldsymbol{J_p}\Delta \boldsymbol{p}] \end{aligned} \tag{16}$$

where $\boldsymbol{J_p} = \frac{\partial \boldsymbol{h}(\boldsymbol{p})}{\partial \boldsymbol{p}} = \{\boldsymbol{J^p_{mwt}}\}$ is the Jacobian matrix (derived in [19, eqs. (4)–(6)]) for differential (1) and algebraic equations (2).

Solutions of (16) provide the system of linear equations for uncertain parameter increments

$$\left[\boldsymbol{J_p^{\mathrm{T}}} \boldsymbol{W} \boldsymbol{J_p}\right] \Delta \boldsymbol{p}\ = \boldsymbol{J_p^{\mathrm{T}}}\ \boldsymbol{W}\,[\boldsymbol{h} - \boldsymbol{h}\,(\boldsymbol{p})] \tag{17}$$

However, the matrix $\boldsymbol{J_p^{\mathrm{T}}} \boldsymbol{W} \boldsymbol{J_p}$ is often ill-conditioned, with eigenvalues spanning several orders of magnitude. To remedy these shortcomings, the approach from [19] was applied.

Additional clarification for the number of analyzed time windows patterns in the nonlinear optimization (14), (15) is necessary. The following typical cases may be identified:

1) Optimization of *very quickly changing parameters* (such as, for example, parameters of *Motor D* of WECC load model—see *Appendix*), where $N'_{\boldsymbol{w}} = 1$ (optimization on the single time window between two consecutive measurement scans).
2) Optimization of *quickly changing parameters* (such as, for example, parameters of the static part of WECC load model—see *Appendix* and [20]–[22]), where the nonlinear optimization can be performed on an hourly basis (for example, for 10 s SCADA refreshing time step, $N'_{\boldsymbol{w}} = 6 \cdot 60 = 360$).
3) Optimization of *slowly changing parameters* (such as, for example, parameters of *Motors A*, *B*, and *C*—see *Appendix* and [20]–[22]), where the nonlinear optimization can be performed on a daily (or longer) basis (for example, for 10 s SCADA refreshing time step, $N'_{\boldsymbol{w}} = 6 \cdot 60 \cdot 24 = 8640$).

V. Transient Analysis

Transient analysis (*TA*) identification is described for *Case 2/4*: SCADA/PMU, the gray-box model (Section II-B).

In cases where the form of the power system dynamic model is known (with uncertain parameters—the gray-box model), the *TA* is performed for time windows ($\boldsymbol{w} \in \mathcal{W}$) as

$$TA_w\,(\boldsymbol{P}_{p,\boldsymbol{w}}, \boldsymbol{P}_{g,\boldsymbol{w}-1})\,,\ \boldsymbol{w}\ =\ 2, 3, \ldots,\ N_{\boldsymbol{w}} \tag{18}$$

subject to a set of disturbances (*Dist*) in the analyzed time interval—integration time ($T_w = [t^{\max}(\boldsymbol{w}-1); t^{\max}(\boldsymbol{w})]$)

$$\sum_{d \in [0; T_w]} Dist_d \tag{19}$$

where a typical disturbance is the total generation increment [from generation forecast, or generation daily profiles, as well as automatic generation control (AGC) settings], determined for the set of AGC participating generators as

$$Dist \equiv \Delta\, \boldsymbol{P}_g = \boldsymbol{P}_g\ (T_w) - \boldsymbol{P}_g\,(0) \tag{20}$$

VI. Interpolation/Extension Schemes

The diffusion maps-based manifold learning algorithm provides a mapping from each measurement sample $\boldsymbol{y}$ ($\boldsymbol{y} \in \{\boldsymbol{Y}(\boldsymbol{m}, \boldsymbol{w}, t);\ \boldsymbol{m} \in \mathcal{M},\ \boldsymbol{w} \in \mathcal{W},\ t\ =\ 1,\ 2,\ \ldots, N_t\} \in \mathbb{R}^N$) in the data space to the K-dimensional reduced space,

represented by elements of eigenvectors (embedding variables) $\boldsymbol{\psi}\,(\boldsymbol{y}) = [\psi_1\ \cdots\ \psi_k \cdots\ \psi_K]^{\mathrm{T}}$—*restriction operator*, where $\boldsymbol{\psi_k} = \psi_k\,(t)$. The manifold ($\boldsymbol{\Omega}$) in the data space ($\boldsymbol{Y}$) is the set of points on $\boldsymbol{\Omega}$, that gets mapped to the corresponding reduced set $\{\boldsymbol{\psi}_k\}$ on the manifold in the reduced space ($\boldsymbol{\Omega}_{\mathrm{red}}$). We can use any interpolation technique to compute $\boldsymbol{y}$ for any other value of $\boldsymbol{\psi}_k$, or $\boldsymbol{y} = \Theta(\boldsymbol{\psi}_k)$—*lifting operator*. We can construct a mapping in the other direction, $\boldsymbol{\psi}_k = \Theta^{-1}\ (\boldsymbol{y})$ (*restriction operator*). We need to conceptually recast changes in measurements from (3), $\mathrm{d}\boldsymbol{y}/\mathrm{d}t = \mathrm{d}\boldsymbol{h}(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{p}, t)/\mathrm{d}t$, into the reduced space as [12]

$$\frac{\mathrm{d}\boldsymbol{\psi}_k}{\mathrm{d}t} = \left(\boldsymbol{J}^{\mathrm{T}}\boldsymbol{J}\right)^{-1}\boldsymbol{J}\frac{\mathrm{d}\boldsymbol{y}}{\mathrm{d}t} \tag{21}$$

where $\boldsymbol{J} = \mathrm{d}\Theta/\mathrm{d}\boldsymbol{\psi}_k$.

Summarizing, for a given value of $\boldsymbol{\psi}_k$, we need a computational method to evaluate $\mathrm{d}\boldsymbol{\psi}_k/\mathrm{d}t$, while we have available a method to compute $\mathrm{d}\boldsymbol{y}/\mathrm{d}t$ (for the black-box and the gray-box models this can be calculated numerically, while for the gray-box models analytical techniques can be brought in, see [19]). To perform this calculation, we execute the following steps:

1) Compute the $\boldsymbol{y}$ on $\boldsymbol{\Omega}_{\mathrm{red}}$ corresponding to the current $\boldsymbol{\psi}_k$ (using any form of interpolation).
2) Compute the change $\mathrm{d}\boldsymbol{y}/\mathrm{d}t$.
3) Compute the change in reduced space $\mathrm{d}\boldsymbol{\psi}_k/\mathrm{d}t$ by (21).

Several possible extensions (in effect, interpolation/extrapolation) schemes can be adopted for calculating the lifting operator (Θ) and restriction operator (Θ^{-1}), as well as $\boldsymbol{J} = \mathrm{d}\Theta/\mathrm{d}\boldsymbol{\psi}_k$ in (21), as shown in the sequel.

A. The Nyström Extension

The Nyström extension is a technique for the numerical approximation of eigenvectors and for obtaining the k-th approximation coordinate ($\hat{\boldsymbol{\psi}}_k$) at an arbitrary data pattern ($\boldsymbol{y}$) as [11]

$$\hat{\boldsymbol{\psi}}_k\ (\boldsymbol{y}) = \frac{1}{N\lambda_k}\ \sum_{i\,=\,1}^{N} W\,(\boldsymbol{y}, \boldsymbol{y}_i)\,\hat{\boldsymbol{\psi}}_k\,(\boldsymbol{y}_i)\,, \tag{22}$$

where the weighting factor from (7) is

$$W\ (\boldsymbol{y}, \boldsymbol{y}_i) = e^{-\left(\frac{\|\boldsymbol{y}-\boldsymbol{y}_i\|}{\varepsilon}\right)^2} \tag{23}$$

and

$$\boldsymbol{y}_i \in \{\boldsymbol{y}_{\boldsymbol{m}}, \boldsymbol{y}_{\boldsymbol{w}}, \boldsymbol{y}_t\} = \left\{\boldsymbol{Y}\,(\boldsymbol{m}, \boldsymbol{w}, t)\,;\ \boldsymbol{m} \in \mathcal{M},\ \boldsymbol{w} \in \mathcal{W},\ t\ =\ 1,\ 2,\ \ldots, N_t\right\} \in \mathbb{R}^N$$

λ_k is k-th eigenvalue;

$\|\boldsymbol{y} - \boldsymbol{y}_i\|$ denotes the Euclidean distances between data points $\boldsymbol{y}$ and $\boldsymbol{y}_i$.

This method allows us to extend an eigenvector computed for a set of sample points $[\hat{\boldsymbol{\psi}}_k(\boldsymbol{y}_i)]$ to an arbitrary point $[(\hat{\boldsymbol{\psi}}_k(\boldsymbol{y})]$ using $W(\boldsymbol{y}, \boldsymbol{y}_i)$ as the interpolation weights. The Jacobian matrix ($\boldsymbol{J}$) in (21), may be obtained by differentiation of (22)—see [12].

B. Radial Basis Functions

Both the *lifting* and *restriction operators* may be obtained by locally interpolating through the radial basis functions.

The corresponding point in the full (data) space, or the lifting operator, $\boldsymbol{y} = \Theta(\boldsymbol{\psi}_k)$, can be generally expressed as [12]

$$\boldsymbol{y}\ = \sum_{i=1}^{NN} \alpha_{ik}\|\boldsymbol{\psi}_k - \boldsymbol{\psi}_i\|^p \tag{24}$$

over the *NN* nearest neighbors of $\boldsymbol{\psi}_k$, where p is an odd integer, while $\|\cdot\|$ denotes the Euclidean distance in the reduced space ($\boldsymbol{\Omega}_{\mathrm{red}}$).

The coefficients α_{ik} are computed as

$$\begin{bmatrix} \alpha_{1k} \\ \alpha_{2k} \\ \vdots \\ \alpha_{NNk} \end{bmatrix} = \Lambda^{-1} \begin{bmatrix} y_{1k} \\ y_{2k} \\ \vdots \\ y_{NNk} \end{bmatrix};\ \Lambda\ (i,j) = \|\boldsymbol{\psi}_i - \boldsymbol{\psi}_j\|;$$

$$i, j\ =\ 1,\ 2,\ \ldots, NN \tag{25}$$

Similarly, the *restriction operator* [$\boldsymbol{\psi}\ = \Theta^{-1}\ (\boldsymbol{y})$] may be expressed in the form

$$\hat{\boldsymbol{\psi}}_k\ (\boldsymbol{y}) = \sum_{i=1}^{NN} \alpha_{ik}\|\boldsymbol{y} - \boldsymbol{y}_i\|^p \tag{26}$$

The Jacobian matrix ($\boldsymbol{J}$) in (21) can be obtained by differentiating (26)—see [12].

C. The Kriging Predictor

In Kriging, the extension of an input function to a new point is performed via a weighted linear combination of the input function at known points. A noticeable feature of the Kriging predictor is that weights may depend on both *distance* (24) *and correlations* between the available samples [23].

The random process (z) with mean zero and covariance is

$$\mathbb{E}\ [z\,(x)\,z\,(y)] = \sigma^2\,\mathcal{R}\,(\theta, x, y) \tag{27}$$

between $z(x)$ and $z(y)$, where σ^2 is the variance of the response, and $\mathcal{R}(\theta, x, y)$ is the correlation model with parameter θ.

Given a set of N inputs $\boldsymbol{y}(\boldsymbol{\psi}_k) = [\boldsymbol{y}_1\ \cdots\ \boldsymbol{y}_i\ \cdots \boldsymbol{y}_N]^{\mathrm{T}}$, with $\boldsymbol{y}_i \in \{\boldsymbol{y}_{\boldsymbol{m}}, \boldsymbol{y}_{\boldsymbol{w}}, \boldsymbol{y}_t\} = \{\boldsymbol{Y}(\boldsymbol{m}, \boldsymbol{w}, t);\ \boldsymbol{m} \in \mathcal{M},\ \boldsymbol{w} \in \mathcal{W},\ t\ =\ 1,\ 2,\ \ldots, N_t\}$ and responses $\boldsymbol{\psi}_k \in [\boldsymbol{\psi}_1\ \cdots\ \boldsymbol{\psi}_k\ \cdots\ \boldsymbol{\psi}_K]^{\mathrm{T}}$, with $\boldsymbol{y}_i = y_i\ (t) \in \mathbb{R}^N$ and $\boldsymbol{\psi}_k = \psi_k\ (t) \in \mathbb{R}^K$. We adopt an optimal prediction model $\hat{\boldsymbol{y}}(\boldsymbol{\psi}_k)$ that expresses the deterministic response $\boldsymbol{y}(\boldsymbol{\psi}_k) \in \mathbb{R}^N$ as a realization of a regression model ($\mathcal{F}$) and random function

$$\hat{\boldsymbol{y}}\,(\boldsymbol{\psi}_k)\ \ =\ \mathcal{F}\,(\boldsymbol{\beta}_k, \boldsymbol{\psi}_k) + \boldsymbol{e}_k\,(\boldsymbol{\psi}_k) \tag{28}$$

where $\boldsymbol{\beta}_k$ is the vector of regression parameters and $\boldsymbol{e}_k(\boldsymbol{\psi}_k)$ is an error of k-th component.

We use a regression model ($\mathcal{F}$) which is a linear combination of P chosen functions ($\boldsymbol{f}_p[\boldsymbol{y}(\boldsymbol{\psi}_k)];\ p\ =\ 1,\ 2,\ \ldots, P$

$$\mathcal{F}\,(\boldsymbol{\beta}_k, \boldsymbol{\psi}_k) = \beta_{1k}\ \boldsymbol{f}_1\,[\boldsymbol{y}\,(\boldsymbol{\psi}_k)] + \beta_{2k}\boldsymbol{f}_2\,[\boldsymbol{y}\,(\boldsymbol{\psi}_k)] + \cdots + \beta_{Pk}\boldsymbol{f}_P\,[\boldsymbol{y}\,(\boldsymbol{\psi}_k)]\ = \boldsymbol{f}_p[\boldsymbol{y}\,(\boldsymbol{\psi}_k)]^{\mathrm{T}}\,\boldsymbol{\beta}_k \tag{29}$$

We have expanded the $(N \times P)$-dimensional measurement matrix as

$$\boldsymbol{F} = \left[\boldsymbol{f}_p\left[\boldsymbol{y}_1\left(\boldsymbol{\psi}_k\right)\right]\ \boldsymbol{f}_p\left[\boldsymbol{y}_2\left(\boldsymbol{\psi}_k\right)\right]\ \cdots \boldsymbol{f}_p\left[\boldsymbol{y}_N\left(\boldsymbol{\psi}_k\right)\right]\right]^{\mathrm{T}} \quad (30)$$

with $\boldsymbol{f}_p[\boldsymbol{y}(\boldsymbol{\psi}_k)]$ defined in (29).

Further, define the matrix $\boldsymbol{R}$ with stochastic-process correlations between measurements, with elements

$$R_{ij} = \mathcal{R}\left(\theta, \boldsymbol{y}_i, \boldsymbol{y}_j\right), \quad (31)$$

$i, j = 1, 2, \ldots, N$ and the vector of correlations between the measurement vector $(\boldsymbol{y}_i)$ and eigenvector $(\boldsymbol{\psi}_k)$ as

$$\boldsymbol{r}_k\left(\boldsymbol{y}_i, \boldsymbol{\psi}_k\right) = \left[\mathcal{R}\left(\theta, \boldsymbol{y}_1, \boldsymbol{\psi}_k\right) \cdots \mathcal{R}\left(\theta, \boldsymbol{y}_N, \boldsymbol{\psi}_k\right)\right]^{\mathrm{T}} \quad (32)$$

Considering the linear predictor $\boldsymbol{c} = \boldsymbol{c}[\boldsymbol{y}(\boldsymbol{\psi}_k)]$ and $\boldsymbol{y}(\boldsymbol{\psi}_k) = [\boldsymbol{y}_1 \cdots \boldsymbol{y}_i \cdots \boldsymbol{y}_N]^{\mathrm{T}}$ defined above, we have

$$\hat{\boldsymbol{y}}\ \left(\boldsymbol{\psi}_k\right) = \boldsymbol{c}^{\mathrm{T}}\ \boldsymbol{y}\left(\boldsymbol{\psi}_k\right) \quad (33)$$

and the error is

$$\begin{aligned}\hat{\boldsymbol{y}}\left(\boldsymbol{\psi}_k\right) - \boldsymbol{y}\ \left(\boldsymbol{\psi}_k\right) &= \boldsymbol{c}^{\mathrm{T}}\ \boldsymbol{\psi}_k - \boldsymbol{y}\left(\boldsymbol{\psi}_k\right) \\ &= \boldsymbol{c}^{\mathrm{T}}\ \left(\boldsymbol{F}\boldsymbol{\beta}_k + \boldsymbol{E}\right) - \left(\boldsymbol{f}[\boldsymbol{y}\left(\boldsymbol{\psi}_k\right)]^{\mathrm{T}}\boldsymbol{\beta}_k + \boldsymbol{e}_k\right)\end{aligned} \quad (34)$$

where $\boldsymbol{E}$ is an error vector of measurements $(\boldsymbol{y}(\boldsymbol{\psi}_k))$.

To keep the predictor unbiased we require that

$$\boldsymbol{F}^{\mathrm{T}}\boldsymbol{c} = \boldsymbol{f}\left(\boldsymbol{y}\left(\boldsymbol{\psi}_k\right)\right) \quad (35)$$

Under this condition, the mean-square error (MSE) of the predictor (34) is [23, eq. (2.11)]

$$\begin{aligned}\mathrm{MSE} &= \mathbb{E}\left[\left(\hat{\boldsymbol{y}}\left(\boldsymbol{\psi}_k\right) - \boldsymbol{y}\left(\boldsymbol{\psi}_k\right)\right)^2\right] \\ &= \mathbb{E}\left[\left(\boldsymbol{c}^{\mathrm{T}}\boldsymbol{E} - \boldsymbol{e}_k\right)^2\right] = \sigma^2\ \left(1 + \boldsymbol{c}^{\mathrm{T}}\boldsymbol{R}\boldsymbol{c} - 2\boldsymbol{c}^{\mathrm{T}}\boldsymbol{r}_k\right)\end{aligned} \quad (36)$$

Solving the optimization criterion (34), subject to the constraint (33) gives the optimal lifting operator, $\boldsymbol{y} = \Theta(\boldsymbol{\psi}_k)$. For solution details, please see [23].

VII. Application

The proposed algorithms for load data classification and reduction, parameter identification, and state extension were tested on a real-world test system (Electric Power Industry of Serbia; a part of the European Network of Transmission System Operators (ENTSO-E) interconnection) with 441 buses, 280 load buses, 655 branches (transmission lines and two/three winding transformers), 78 production units [synchronous generators (SGs), double fed induction generators (DFIGs) and direct drive induction generators (DDSGs)], with automatic voltage regulators (AVRs) and turbine models, where SGs, DFIGs, DDSGs, AVRs, and turbines (steam, hydro, wind, and solar) are modeled by different numbers of equations. The dynamic model has a total of 1027 state variables. Part of the generation units (a total of 14) participates in AGC, depending on their participation factors, covering an imbalance between total load and generation, obtained from daily profiles for loads (four load types are available) and generations (thermal, hydro, wind, and solar units are available), respectively. The transient analysis (*TA* in Section V) is performed by PSS/E software (version 33.5.2) [24], where the WECC load is modeled by the user-defined CMLDBLU1 model.

TABLE I
Largest Eigenvalues in Time Windows Axis and Times Axis

Number	Time windows axis	Time axis
0	1	1
1	3.3074e-04	5.5880e-06
2	6.9695e-08	6.0575e-08
3	1.8520e-08	3.7828e-08
4	1.1266e-08	2.1915e-08
5	1.0112e-08	1.8072e-08
6	8.7365e-09	1.1526e-08
7	8.3219e-09	1.0056e-08
8	6.2550e-09	9.3101e-09
9	5.6635e-09	8.1815e-09

The generation increments (for 14 participating units in AGC) in time windows as described in Section V are used as a disturbance for transient analysis [$Dist$ in (20)].

A. Basic Results

Basic results are shown for *Case 2: SCADA measurements, the gray-box model* (see Section II-B).

We generate a three-dimensional set of measurements ($\mathcal{M}$) at the WECC connection point 110 kV (bus 34390, JBOGAT5): active power (P_{lf}), reactive power (Q_{lf}) and bus voltage magnitude (V_{lf})—see Fig. 1 for clarification. From these values may be calculated values in the load bus: P_{load}, Q_{load}, and V_{load}. Without loss of generality (the proposed methodology similarly can be applied to other dynamical models in Fig. 10), the obtained results are shown on the example of the single-phase *Motor D*, where active and reactive loads of *Motor D* are: P_D and Q_D, respectively—see *Appendix* and Fig. 10 (*Motor D* is highlighted by the red rectangle) for clarification. For each $\boldsymbol{m} \in \mathcal{M}$ and $\boldsymbol{w} \in \mathcal{W}$, we observe the trajectory of the WECC dynamic load at the *Motor D* level sampled every 10 s ($6\times60\times24-1 = 8639$ time windows), with $N_t = 172$ points of transient analysis for the time of the transients 10 s. We collect all of the trajectories into a single 3-D measurement tensor ($\boldsymbol{Y}$), where $\boldsymbol{Y} \in \mathbb{R}^{3\times8639\times172}$, with the trajectories of the system variables $\boldsymbol{y}(\boldsymbol{m}, \boldsymbol{w}, t)$, $t = 1, 2, \ldots, N_t$, where t denotes the time samples. Surfaces of the tensor $\boldsymbol{Y}$ components (active powers, reactive powers, and bus voltages) are shown in Fig. 1.

Pattern classification by manifold learning, which involves diffusion geometry (diffusion maps) with data-driven partition trees is based on the methodology from [14].

The number of largest eigenvalues (λ_k, sorted in descending order) calculated for the diffusion map embedding of the samples in reduced space (9) is decided by the user-selected parameter (which also strongly affects the total calculation time). In Table I, we show the $K = 9$ largest eigenvalues of the row stochastic matrix ($\boldsymbol{A}$). From Table I, we conclude that eigenvalues decrease

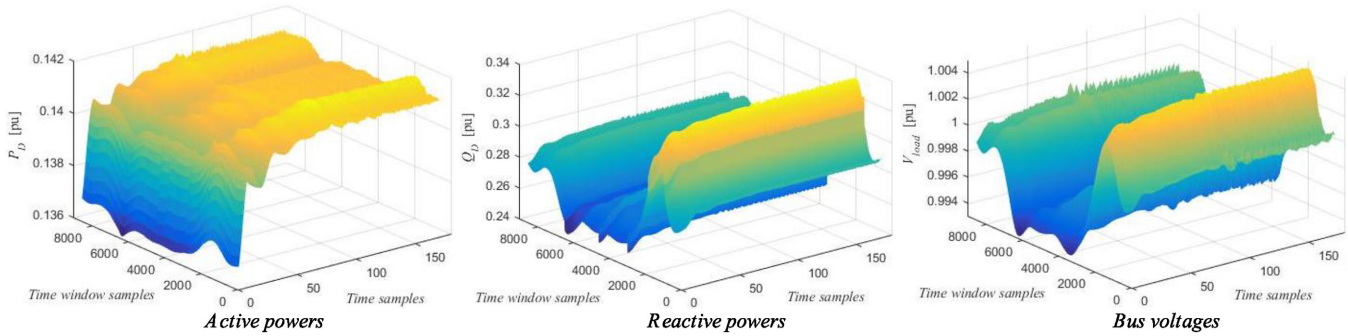

Fig. 1. Measurement tensor ($\boldsymbol{Y}$).

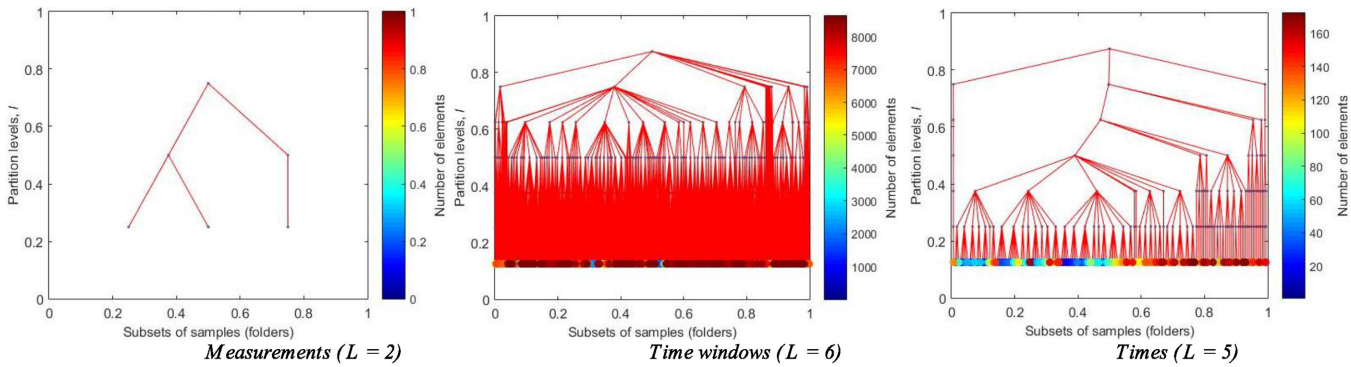

Fig. 2. Partition trees in different tensor axes.

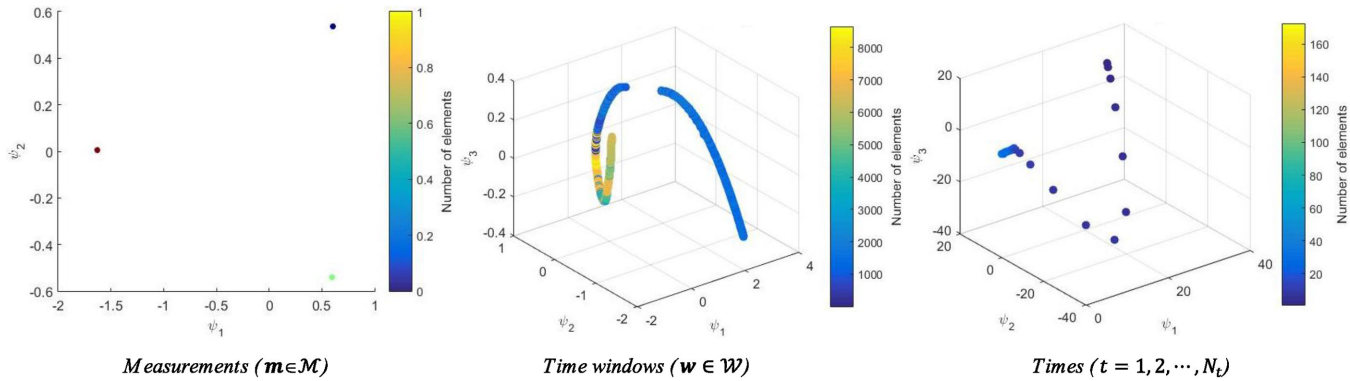

Fig. 3. Dominant embedding (elements of eigenvectors) in different tensor axes.

approximately exponentially, suggesting that only a small number of the largest eigenvalues need to be calculated. In [10, [25] it was demonstrated that the truncation down to ∼10% still enables appropriate reconstruction.

The exponential decay of eigenvalues is reminiscent of a similar distribution for the eigenvalue spectrum of the Fisher Information Matrix (FIM) in physics-based models of power systems (often called "sloppy" in the physics literature) [26]. In both cases, small eigenvalues indicate irrelevance of the associated degrees of freedom, and the two approaches are dual to one another. Here we use eigenvalues of the diffusion map embedding to identify low-dimensional embeddings of the model in a data-driven way. Elsewhere, we have used the FIM to perform parameter reduction of physics-based models [27].

Partition trees and dominant embedding (elements of eigenvectors) (9), obtained by the diffusion maps-based manifold learning algorithm (Section III), in different tensor axes are shown in Fig. 2 and Fig. 3, respectively.

From Fig. 1 and Fig. 3 the following conclusions may be derived:

1) The discontinuity in the plot 'Time windows ($\boldsymbol{w}$)' for reactive power is generated by the stalling effect and voltage-reactive power dependence—see [13, Fig. 1] and the plots for reactive powers and bus voltages in Fig. 1.

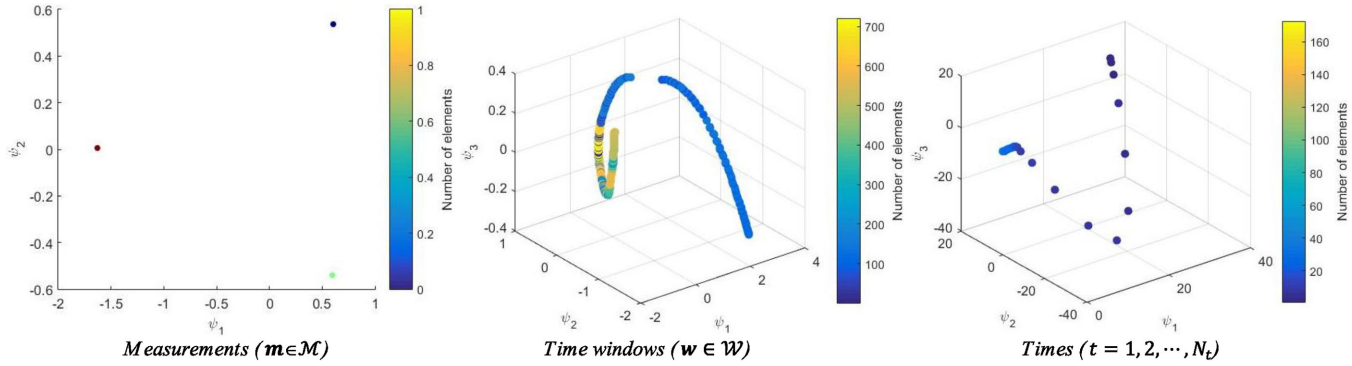

Fig. 4. Dominant embedding (elements of eigenvectors) in different tensor axes (reduced tensor).

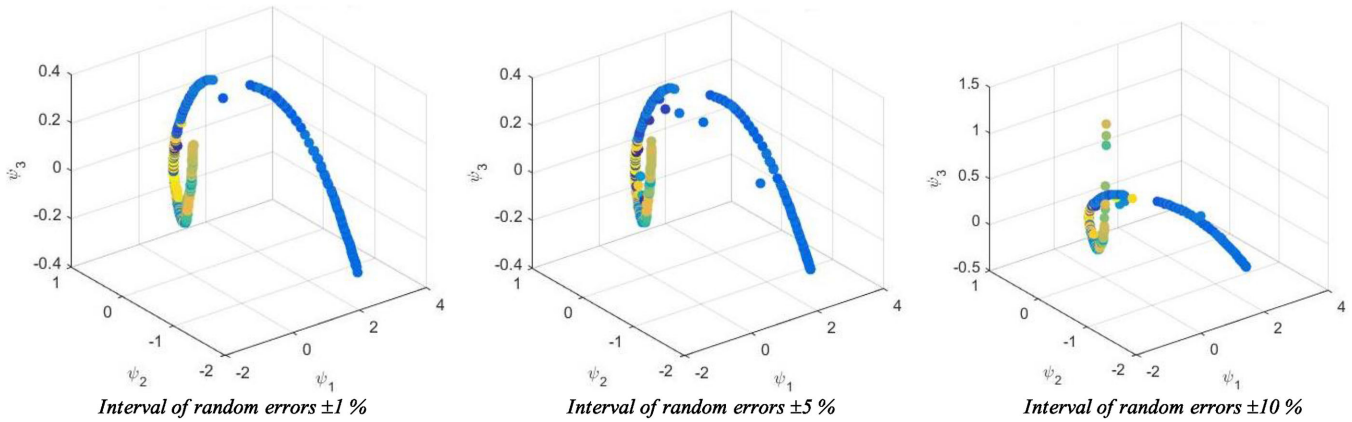

Fig. 5. Influence of 20 bad transients data ($t = 1, 2, \ldots, N_t$) of reduced tensor in time windows to dominant embedding (elements of eigenvectors).

2) Transient analysis reaches the (quasi) steady-state condition for approximately 10 s (see Fig. 1), so it is not necessary to increase the integration time.
3) In cases where the power system is subject to small daily load/generation variations (no major disturbances), the elements of the eigenvector also vary just slightly (see Fig. 3).
4) With this in mind, a good state prediction based on the elements of eigenvector prediction is to be expected.
5) Looking at 'Time windows ($\boldsymbol{w} \in \mathcal{W}$)' the plot in Fig. 3, it is expected that a very similar plot can be obtained by reducing the size of the time windows of the input (measurement) tensor. Elements of the eigenvectors along different tensor axes for the reduced tensor [*Motor D* is sampled in every 2 min in one day (24 hours), or $30\times24-1 = 719$ time windows— $\boldsymbol{Y} \in \mathbb{R}^{3\times719\times172}$] are shown in Fig. 4. When comparing plots from Fig. 3 and Fig. 4, we can conclude that the reduced measurement tensor provides enough information for state prediction (see Section VII-C).

The proposed method may be used for the identification and filtering of bad data. In the reduced tensor ($\boldsymbol{Y} \in \mathbb{R}^{3\times719\times172}$) are assumed 20 bad transients in the time windows axis ($\boldsymbol{y}_{\boldsymbol{w}}$), with three characteristic intervals of random errors: ±1 %, ±5 %, and ±10 %. Dominant embedding, or elements of the eigenvectors of the row stochastic matrix ($\boldsymbol{A}$) in the time windows axis are shown in Fig. 5 (note that the dominant embedding in the measurements axis and the time axis are the same as in Fig. 4). From the presented results, the bad elements of the eigenvector and their sources by participation factors may be identified.

The computation time for manifold learning with reduced tensor is approximately 7 s.[1]

B. Parameter Identification

The proposed algorithm for parameter identification (Section IV) is verified on the example of uncertain parameters for the single-phase *Motor D* [22], see Fig. A1:

X_{stall} – stall reactance, in [pu];
T_{th} – thermal time constant, in [s];
θ_{2t} – thermal protection trip completion level (in [pu] temperature).

Unavailable real-time (typically PMU-based) measurements in point where the WECC load model is connected, for on-line parameter estimation, are replaced with 50 random variations of the above parameter set in range ±10 % (from values given in [20]–[22]: $X_{stall} = 0.1$, $T_{th} = 15$, and $\theta_{2t} = 1.9$). These parameter sets are used to calculate the trajectories of the measured variables $[y(t)]$. For three available measurements ($\boldsymbol{m} \in \mathcal{M} = \{P_D, Q_D, V_{load}\}$) and *one time window* ($\boldsymbol{w} \in \mathcal{W}$) with

[1]Intel(R) Core(TM) i7-6860HQ CPU @ 2.70 GHz, 64-bit Operating System, 32 GB RAM.

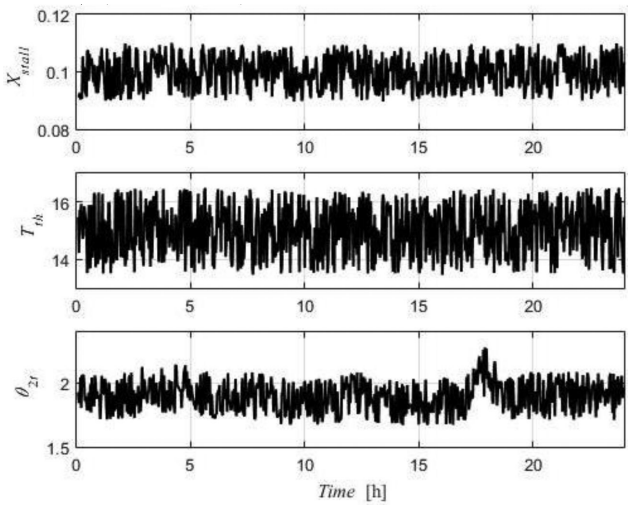

Fig. 6. Optimized parameters of *Motor D* for the analyzed daily profile.

$N_p = 50$ parameter variations we generate the measurement tensor $\boldsymbol{Y} \in \mathbb{R}^{3\times 50\times 172}$ with trajectories $Y(\boldsymbol{m}, \boldsymbol{w}, t)$, where for *very quickly changing parameter* case (from Section IV) are: $N'_{\boldsymbol{m}} = 3$, $N'_{\boldsymbol{w}} = 1 \cdot N_p = 50$, and $N_t = 172$.

Optimized parameters of *Motor D* for the analyzed daily load profile are shown in Fig. 6. In this figure, the stalling effect clearly can be identified in the change of θ_{2t} between the 17th and 19th hours.

The computation time for parameter identification is approximately 1.2 s/time window ($\boldsymbol{w}$).

C. State Prediction

For state prediction, input ($\boldsymbol{y}$) and output ($\boldsymbol{\psi}_k$) for *restriction operator*, $\boldsymbol{\psi}_k = \Theta^{-1}(\boldsymbol{y})$ (Section VI), are composed of a set of measurements (tensor $\boldsymbol{Y}$, shown in Fig. 1 and used for the calculation in Section VII-A) and the calculated embedding {eigenvalues (λ_k) and elements of eigenvectors [$\psi_k(t)$]; $k = 1, 2, \ldots, K$, shown in Fig. 7 for four ($k = 1; 2; 3; 4$) dominant eigenvalues}, respectively.

To generate the Kriging model (Section VI-C), we use the DACE toolbox in Matlab [23].

For state prediction, the reduced measurement tensor ($\boldsymbol{Y} \in \mathbb{R}^{3\times 719\times 172}$, Section VII-A) is used (to reduce the calculation time further, only a selected number of time windows ($\boldsymbol{w} \in \mathcal{W}$) trajectories may be used—for example, every 20th). Input measurement vector for state (measurement) prediction (22) and eigenvector for dominant eigenvalue ($k = 1$), $\boldsymbol{\psi}_1$, for DACE toolbox are prepared as:

$$\boldsymbol{y}(\boldsymbol{m}, \boldsymbol{w}, t)^{\mathrm{DACE}} = \begin{bmatrix} \boldsymbol{y}(\boldsymbol{m}, 1, t) \\ \vdots \\ \boldsymbol{y}(\boldsymbol{m}, \boldsymbol{w}, t) \\ \vdots \\ \boldsymbol{y}(\boldsymbol{m}, N_{\boldsymbol{w}}, t) \end{bmatrix}$$

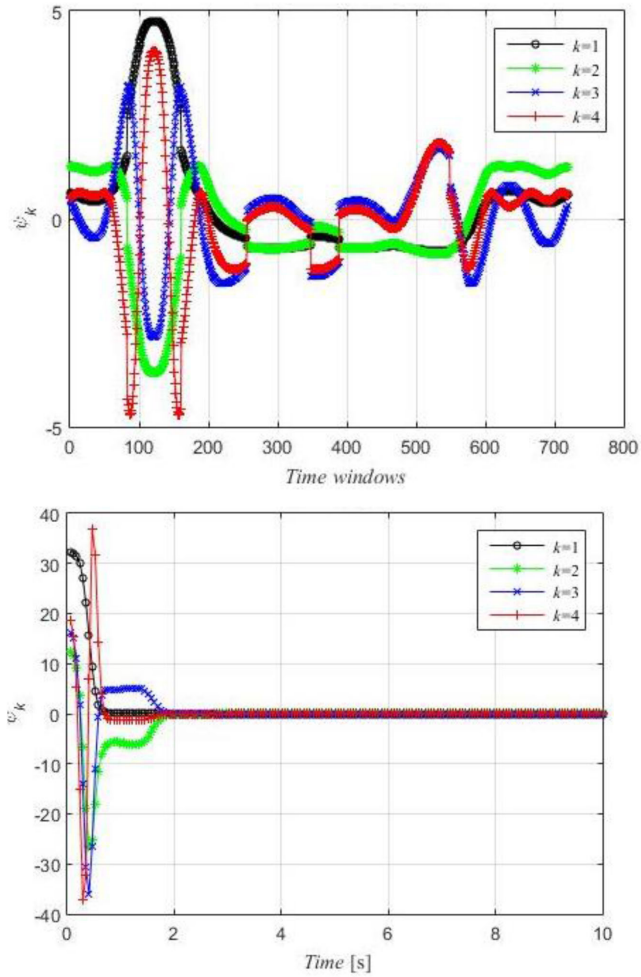

Fig. 7. Elements of the eigenvector for prediction model [*restriction operator*, $\boldsymbol{\psi}_k = \psi_k(t) = \Theta^{-1}(\boldsymbol{y})$] in time windows axis (top panel) and time axis (bottom panel).

$$\boldsymbol{\psi}_1^{\mathrm{DACE}} = \begin{bmatrix} \boldsymbol{\psi}_1 \\ \vdots \\ \boldsymbol{\psi}_1 \\ \vdots \\ \boldsymbol{\psi}_1 \end{bmatrix}$$

where:

$$\boldsymbol{m} \in \mathcal{M} = \{P_D, Q_D, V_{load}\}$$

$$\boldsymbol{y}(\boldsymbol{m}, \boldsymbol{w}, t) = \begin{bmatrix} y(\boldsymbol{m}, \boldsymbol{w}, 1) \\ \vdots \\ y(\boldsymbol{m}, \boldsymbol{w}, t) \\ \vdots \\ y(\boldsymbol{m}, \boldsymbol{w}, N_t) \end{bmatrix}; \; \boldsymbol{y}(\boldsymbol{m}, \boldsymbol{w}, t) \in \{\boldsymbol{Y}\}$$

$$\boldsymbol{\psi}_1 = \begin{bmatrix} \psi_1(1) \\ \psi_1(2) \\ \vdots \\ \psi_1(N_t) \end{bmatrix}$$

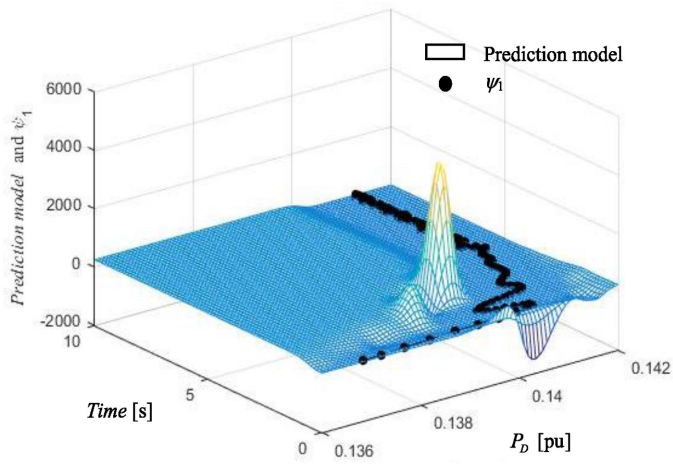

Fig. 8. State (active power transients) prediction model [*lifting operator*, $\boldsymbol{y}(P_D, \boldsymbol{w}, t) = \Theta(\boldsymbol{\psi}_1)$] for dominant eigenvalue (λ_1), $\boldsymbol{\psi}_1 = \psi_1(t)$.

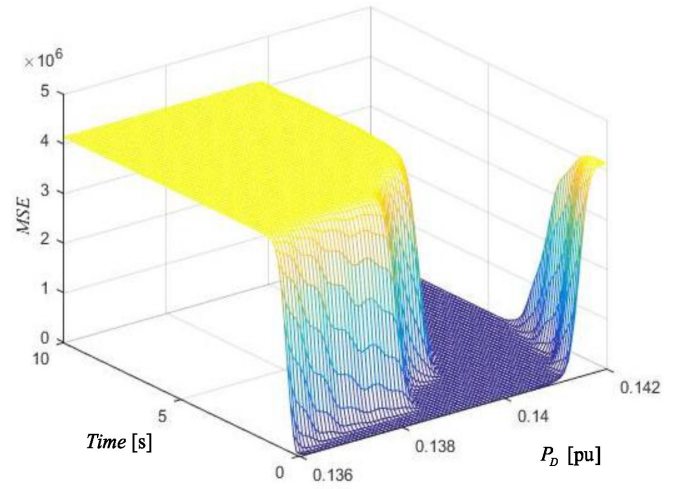

Fig. 9. MSE of state [active power transients, $\boldsymbol{y}(P_D, \boldsymbol{w}, t)$] prediction model for the dominant eigenvalue (λ_1), $\boldsymbol{\psi}_1 = \psi_1(t)$.

In the two-column input measurement matrix requested for the DACE toolbox, the first column is defined above $(N_{\boldsymbol{w}} \cdot N_t)$-dimensional measurement column-vector $\boldsymbol{y}(\boldsymbol{m}, \boldsymbol{w}, t)^{\mathrm{DACE}}$, while the second column is composed of the corresponding repeated time points $(t = 1, 2, \ldots, N_t)$ for the analyzed time windows ($\boldsymbol{w} \in \mathcal{W}$). Similarly, $\boldsymbol{\psi}_1^{\mathrm{DACE}}$ is $(N_{\boldsymbol{w}} \cdot N_t)$-dimensional vector.

The state (measurement) prediction model is shown on the 'Active powers' example in Fig. 1 [$\boldsymbol{y}(P_D, \boldsymbol{w}, t)$], where this surface is divided into an (80×80)-dimensional meshed grid.

Results for the prediction model obtained from the active power measurement, $\boldsymbol{y}(P_D, \boldsymbol{w}, t)$ and embedding (elements of the eigenvector for the dominant eigenvalue, $k = 1$), $\boldsymbol{\psi}_1 = \psi_1(t)$, are shown in Fig. 8, while the MSE is shown in Fig. 9.

The computation time for the state prediction is approximately 27 s.

From Fig. 9 we can conclude that in the corridor with time responses (first plot in Fig. 1) we have very small MSE, while in uncovered areas with measurement points, MSE increases drastically (as expected).

VIII. Conclusion

The development of mathematical techniques that operate directly on observations, or measurements (data-driven approaches) bypasses the need to precisely select variables and parameters, and to derive accurate equations in closed form (gray-box approach). Initially unorganized measurement data are classified along the dimensions of measurements (inputs), state variables (time windows in our case), parameter settings (inputs), and time snapshot values (depending on the specified tensor), and iteratively lead to the construction of an informed metric for each type of variation.

The manifold learning algorithm that we describe has been tested in black-box and gray-box settings in a large power system. Data classification and reduction have been illustrated on an example of transient dynamics involving a widely used load model. We believe that control design based on this methodology is an interesting avenue to be explored, for example by considering a three-dimensional tensor for each controller output consisting of initial conditions, parameters, and the time axis. The results thus far are encouraging and point toward the need to blend general big-data procedures like manifold learning with power system specific customizations and extensions.

Appendix
WECC Load Dynamic Model

The WECC load dynamic model (shown in Fig. 10) is represented by DAEs, as in [20]–[22] and therein references.

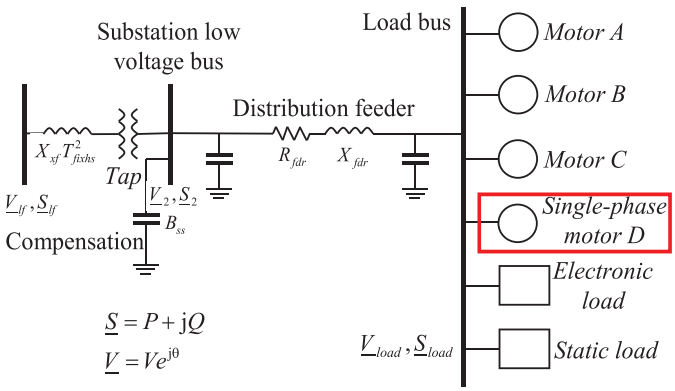

Fig. 10. WECC load dynamic model.

References

[1] S. Lafon, Y. Keller, and R. R. Coifman, "Fusion and multicue matching by diffusion maps," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 11, pp. 1784–1797, Nov. 2006.

[2] X. He, Q. Ai, R. C. Qiu, W. Huang, L. Piao, and H. Liu, "A big data architecture design for smart grids based on random matrix theory," *IEEE Trans. Smart Grid*, vol. 8, no. 2, pp. 674–686, Mar. 2017.

[3] Y. Weng, A. Kumar, M. B. Saleem, and B. Zhang, "Big data and deep learning platform for terabyte-scale renewable datasets," in *Proc. Power Syst. Comp. Conf.*, Dublin, Ireland, Jun. 11–15, 2018, pp. 1–7.

[4] A. Y. Lokhov, M. Vuffray, D. Shemetov, D. Deka, and M. Chertkov, "Online learning of power transmission dynamics," in *Proc. Power Syst. Comp. Conf.*, Dublin, Ireland, Jun. 11–15, 2018, pp. 1–7.
[5] D. Cao *et al.*, "Design and application of big data platform architecture for typical scenarios of power system," in *Proc. IEEE PES Gen. Meeting*, Portland, OR, Aug. 5–9, 2018, pp. 1–5.
[6] Y. Yu, T. Y. Ji, M. S. Li, and Q. H. Wu, "Short-term load forecasting using deep belief network with empirical mode decomposition and local predictor," in *Proc. IEEE PES Gen. Meeting*, Portland, Aug. 5–9, 2018, pp. 1–5.
[7] R. R. Coifman and S. Lafon, "Diffusion maps," *Appl. Comput. Harmonic Anal.*, vol. 21, no. 1, pp. 5–30, Jul. 2006.
[8] M. Belkin and P. Niyogi, "Laplacian Eigenmaps for dimensionality reduction and data representation," *Neural Comput.*, vol. 6, no. 15, pp. 1373–1396, Jun. 2003.
[9] C. M. C. Arvizu and A. R. Messina, "Dimensionality reduction in transient simulations: A diffusion maps approach," *IEEE Trans. Power Syst.*, vol. 31, no. 5, pp. 2379–2389, Oct. 2016.
[10] A. M. Stanković, A. A. Sarić, A. T. Sarić, and M. K. Transtrum, "Interleaving physics- and data-driven models for power system transient dynamics," in *Proc. 21th Power Syst. Comput. Conf.*, Session: Physics-Informed Machine Learning, Porto, Portugal, Jun. 29–Jul. 3, 2020, pp. 1–7.
[11] C. Fowlkes, S. Belongie, F. Chung, and J. Malik, "Spectral grouping using Nyström method," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 2, pp. 214–225, Feb. 2004.
[12] E. Chiavazzo, C. W. Gear, C. J. Dsilva, N. Rabin, and I. G. Kevrekidis, "Reduced models in chemical kinetics via nonlinear data-mining," *Processes*, vol. 2, no. 1, pp. 112–140, 2014.
[13] A. T. Sarić, M. K. Transtrum, and A. M. Stanković, "Data classification and parameter identification in power systems by manifold learning," in *Proc. IEEE PowerTECH Conf.*, Milano, Italy, Jun. 23–27, 2019, pp. 1–6.
[14] O. Yair, R. Talmon, R. R. Coifman, and I. G. Kevrekidis, "Reconstruction of normal forms by learning informed observation geometries from data," *Proc. Nat. Acad. Sci. United States Amer.*, vol. 114, no. 38, pp. E7865–E7874, Aug. 2017.
[15] D. W. Sroczynski, O. Yair, R. Talmon, and I. G. Kevrekidis, "Data-driven evolution equation reconstruction for parameter-dependent nonlinear dynamical systems," *Isr. J. Chem.*, vol. 58, pp. 1–9, Apr. 2018.
[16] J. I. Ankenman, "Geometry and analysis of dual networks on questionnaires," Ph.D. dissertation, Yale Univ., New Haven, CT, USA, 2014.
[17] G. Mishne *et al.*, "Hierarchical coupled-geometry analysis for neuronal structure and activity pattern discovery," *IEEE J. Sel. Top. Signal Process.*, vol. 10, no. 7, pp. 1238–1253, Oct. 2016.
[18] W. E. Leeb, "Topics in metric approximation," Ph.D. dissertation, Yale Univ., New Haven, CT, USA, 2015.
[19] A. T. Sarić, M. K. Transtrum, and A. M. Stanković, "Data-driven dynamic equivalents for power system areas from boundary measurements," *IEEE Trans. Power Syst.*, vol. 34, no. 1, pp. 360–370, Jan. 2019.
[20] D. Kosterev *et al.*, "Load modeling in power system studies: WECC progress update," in *Proc. IEEE PES Gen. Meeting*, Pittsburgh, PA, 2008, pp. 1–8.WECC composite load model specifications 01-27-2015. [Online] Available: https://www.wecc.biz/Reliability/WECC Composite Load Model Specifications 01-27-2015.docx
[21] G. Thoman and J. Senthil, "A simple CMLD initialization (and comparison of PSSE model initialization with PSLF results)," 2017. [Online] Available: https://www.wecc.biz/Administrative/InitializeExample_051516.docx
[22] K. Zhang, H. Zhu, and S. Guo, "Dependency analysis and improved parameter estimation for dynamic composite load modeling," *IEEE Trans. Power Syst.*, vol. 32, no. 4, pp. 3287–3298, Jul. 2017.
[23] S. N. Lophaven, H. B. Nielsen, and J. SΘndergaard, "DACE (Design and Analysis of Computer Experiments) Toolbox," Version 2.0, Aug. 1, 2002. [Online] Available: http://www2.imm.dtu.dk/projects/dace/
[24] "PSSE – high-performance transmission planning and analysis software," Siemens AG, 2019, Accessed: Mar. 1, 2018. [Online]. Available: https://new.siemens.com/global/en/products/energy/services/transmission-distribution-smart-grid/consulting-and-planning/pss-software/pss-e.html
[25] S. L. Brunton and J. N. Kutz, *Data-Driven Science and Engineering: Machine Learning, Dynamical Systems, and Control*. Cambridge, U.K.: Cambridge Univ. Press, 2019.
[26] M. K. Transtrum, A. T. Sarić, and A. M. Stanković, "Information geometry approach to verification of dynamic models in power systems," *IEEE Trans. Power Syst.*, vol. 33, no. 1, pp. 440–450, Jan. 2018.
[27] M. K. Transtrum, A. T. Sarić, and A. M. Stanković, "Measurement-directed reduction of dynamic models in power systems," *IEEE Trans. Power Syst.*, vol. 32, no. 3, pp. 2243–2253, May 2017.

Andrija T. Sarić (Member, IEEE) was born in 1962. He received the B.Sc., M.Sc., and Ph.D. degrees in electrical engineering from the University of Belgrade, Belgrade, Serbia, in 1988, 1992, and 1997, respectively. He is currently a Professor of electrical engineering with the Faculty of Technical Sciences, University of Novi Sad, Novi Sad, Serbia. His main research interests include power system analysis, optimization, and planning, as well as the application of artificial intelligence methods. He is an Associate Editor for the IEEE TRANSACTIONS ON POWER SYSTEMS.

Mark K. Transtrum received the Ph.D. degree in physics from Cornell University, Ithaca, NY, USA, in 2011. He then studied computational biology as a Postdoctoral Fellow from MD Anderson Cancer Center, Houston, TX, USA. He has been with Brigham Young University, Provo, UT, USA, since 2013, and is currently an Associate Professor of physics and astronomy. His research interests include representations of a variety of complex systems including power systems, systems biology, materials science, and neuroscience.

Aleksandar M. Stanković (Fellow, IEEE) received the Ph.D. degree in electrical engineering from the Massachusetts Institute of Technology, Cambridge, MA, USA, in 1993. He is currently an A.H. Howell Professor with Tufts University, Medford, MA, USA. He was with Northeastern University, Boston, MA, USA, from 1993 to 2010. He has held visiting positions with the United Technologies Research Center, and with L'Universite de Paris-Sud, Orsay, France, and Supelec, Gif-sur-Yvette, France. He is a Co-Editor of book series on power electronics and power systems for Springer. He was an Associate Editor for the IEEE TRANSACTIONS ON POWER SYSTEMS, IEEE TRANSACTIONS ON SMART GRID, and IEEE TRANSACTIONS ON CONTROL SYSTEM TECHNOLOGY for more than 20 years.