Integration of Physics- and Data-Driven Power System Models in Transient Analysis After Major Disturbances

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

***Abstract*—The article explores the analysis of transient phenomena in large-scale power systems subjected to major disturbances from the aspect of interleaving, coordinating, and refining physics- and data-driven models. Major disturbances can lead to cascading failures and ultimately to the partial power system blackout. Our primary interest is in a framework that would enable coordinated and seamlessly integrated use of the two types of models in engineered systems. Parts of this framework include: 1) optimized compressed sensing, 2) customized finite-dimensional approximations of the Koopman operator, and 3) gray-box integration of physics-driven (equation-based) and data-driven (deep neural network-based) models. The proposed three-stage procedure is applied to the transient stability analysis on the multimachine benchmark example of a 441-bus real-world test system, where the results are shown for a synchronous generator with local measurements in the connection point.**

***Index Terms*—Compressed sensing, deep learning, dynamic model, Koopman modes, neural network, nonlinear dynamics, power system, system identification.**

NOMENCLATURE

The main symbols used throughout the article are following:

Parameters

r	Rank of the reduced model (selected truncation).
N	Number of points.
t_{end}	Total time for transient analysis.
Δt	Time-step.

Variables, Vectors, Matrices, Sets, and Tensors

$\boldsymbol{A}$	Best-fit linear matrix operator for prediction $\boldsymbol{x}^k \rightarrow \boldsymbol{x}^{k+1}$.
$\boldsymbol{p}$	Vector of parameters.
$\boldsymbol{s}$	Sparse signal in compressed sensing algorithm.
t	Time scalar variable.
$\boldsymbol{v}$	Vector of Koopman modes.
$\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}$	Vector of state, measured, and algebraic variables, respectively.
$\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{Z}$	Training tensors of state, measured, and algebraic variables, respectively.
$\boldsymbol{W}$	Extended vector of state variables, algebraic variables, and calculated measurements.
$\boldsymbol{\Psi}, \boldsymbol{\Phi}$	Transform basis and measurement matrix in compressed sensing algorithm, respectively.
$\boldsymbol{g}$	Vector-valued Hilbert space of measurement function on $\boldsymbol{x}, \boldsymbol{y} = \boldsymbol{g}(\boldsymbol{x})$.
$\boldsymbol{F}_t$	Transition function of states for incremental time-step in state-space.
$\mathcal{K}_t$	Koopman operator of states with time t flow map.
λ	Eigenvalue.
ω	Imaginary part of eigenvalue.
φ	Koopman eigenfunction.
$\mathcal{M}, \mathcal{W}$	Sets of measurements and initial conditions, repectively.

Indices

com	Compressed signal (upper index).
e	Expanded model (lower index).
k	Time instant (upper index).
m	Measurement (lower index).
p	Number of linear signals in compressed sensing algorithm (upper index).
p	Practical model (lower index).
t	Time coordinate (lower index).
$\boldsymbol{x}, \boldsymbol{z}$	State and algebraic variables (lower index).
w	Initial condition in training set (lower index).
0	Initial condition (lower and upper indices).

Abbreviations

DAE	Differential and algebraic equations.
DNN	Deep neural networks.
eDMD	Extended dynamic mode decomposition.
EMS	Energy management system.
MBAM	Manifold boundary approximation method.
PoC	Point of connection.

Manuscript received September 7, 2021; revised December 17, 2021; accepted January 30, 2022. This work was supported in part by the National Science Foundation (NSF) under Grant ECCS-1710944, in part by the CURENT Engineering Research Center of the National Science Foundation and the Department of Energy under National Science Foundation (NSF) under Grant EEC-1041877, and in part by ONR under Grant N00014-16-1-3028. *(Corresponding author: Aleksandar M. Stanković.)*

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

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

Andrija T. Sarić is with the Department for Power, Electronic and Communication Engineering of the Faculty of Technical Sciences, University of Novi Sad, 21102 Novi Sad, Serbia (e-mail: asaric@uns.ac.rs).

Digital Object Identifier 10.1109/JSYST.2022.3150237

SG	Synchronous generator.
TSA	Transient stability analysis.

I. INTRODUCTION

IN MANY engineered systems, including electricity grids, there is a strong, traditional preference for physics-based models. The human-interpretable state spaces and parameter spaces of these models convey physical insights into system behavior, model transferability, and controller design. With the advent of high-performance computing, models have grown in size and complexity. However, large unwieldy models, notwithstanding their apparent detail, are unable to replicate many actual large-scale events, such as blackouts. Sources of difficulties include component variations (e.g., loads and renewable sources); models that exist only in the form of tabulated data or computer code; undocumented alterations of components (e.g., generator exciters), and controller settings. To counterbalance these issues, there is a growing interest in developing mathematical techniques that operate directly on observations or measurements (i.e., data-driven methods) [1]–[3].

These methods have the potential to eliminate some of the fidelity issues with existing physics-based (equation-based) models for different static and dynamic studies in power systems. In the available literature, both data-based and hybrid (physics- and data-based) approaches have been applied to the problems of state estimation [4], power flow [5], optimal power flow [6], angle stability [7], and frequency stability assessment and control [8]. In these references, different machine learning approaches have been used for the data-driven part of models.

Transient stability analysis (TSA), however, is much more complex, since existing approaches for static problems cannot be directly applied. Thus, gray-box methods integrating physics- and data-driven power system models have not been studied as much. Some papers, such as [9], [10], deal with hybrid methods, but only use the analytical part in the sample reduction process, before becoming fully data-driven in the application phase. In contrast, we retain equations even in the online phase, maximizing their contribution. Another advantage of our approach is that the equations we use convey a physical meaning of the system states, instead of compacting them into behavior-summary functions, for instance, Lyapunov function, as in [11].

TSA (and other dynamic analyses, such as dynamic voltage stability, angle stability, etc.) must be performed in real-time (specified as the Analysis Tool 7 in [12]) by the transmission system operators. It is performed by the energy management system (EMS) for the normal operating condition and a list of critical contingency cases, to provide adequate situational awareness and to detect the transient stability criteria violations. Additionally, these results are used to prepare the appropriate remedial action schemes to prevent dynamic instability [13]. The absence of such actions can result in cascading failures and for critical cases in partial blackout(s) [14]. Traditionally, the TSA is a part of EMS, performed using the physics-based models with a user-defined level of details. In our context, such a model is named a *practical model*.

However, the discrepancy between the practical model and measured data always exists (irrespective of the complexity of the practical model). An additional problem is that many state variables are not electric (for example, for turbines), and obtaining them is expensive for real-time application. Furthermore, some typically measured electric quantities are indirectly related to the state and algebraic variables (such as active/reactive power flow and injection and current measurements). Based on these conclusions, our objective is to formulate a hybrid (gray-box) *expanded model*, by extending the practical model with a data-driven part (DNN-based) to fit the discrepancy between the two models. This implies that the practical model can be of lower order (reduced), to cover only the dominant system dynamics. Another motivation for model reduction comes from the fact that much of that data is locally correlated and there are reasons to believe that a considerable portion of the data lies on a manifold of much lower dimension [1]. The effectiveness of low-order, often physics-derived models ("dynamic equivalents") is a testament to this potential reduction.

We envision an integrated workflow and data flow for TSA as a three-stage process:

1) an existing (physics-based) model is tested for practical identifiability using tools from computational information geometry;
2) their states and relationships with measurements are extracted;
3) a data-driven model is derived for the same collections of states using tools from machine learning and Koopman operator theory.

In this article, we describe a coordinated framework for interleaving physics- and data-driven models that allow for iterative improvements in both classes. We review relevant concepts (information geometry, machine learning, compressed sensing, and Koopman operator approximations) within the overall workflow. Out of the rich literature on each of these subjects and their power system applications, we comment only on references that are of immediate relevance for our development. Previous work has demonstrated that information geometry is a suitable framework for addressing above steps 1 and 2 [3]. The manifold boundary approximation method (MBAM) reliably extracts a physics-based (practical) model that is identifiable from a given measurement structure (human-interpretable part in the gray-box model). The proposed workflow and data flow retain the traditional focus on physical states, which is justified by measuring quantities (such as voltages, currents, and power flows) for which there are a priori known physical relationships. In this work, we ask if there exists an even better (expanded) model to be built on top of the earlier iterations of MBAM. Our main tools in searching for more predictive models with a given set of states are deep neural networks (DNN) [15], which have a proven record in capturing many different types of dynamics. Note, however, that this is not the only option and other regression models could also be of interest.

One key tool in deriving dynamical models from data is the approximation by the Koopman modes. It has been successfully used in several power system applications ranging from TSA to oscillations [19]–[23].

The main contributions of the proposed method are following.

1) The proposed workflow and data flow for TSA in power systems integrating physics- (practical model) and data-driven dynamic models (representing the difference in transient responses between the measurements and calculations from the practical model).
2) The proposed DNN-based architecture for integrated gray-box and two-class model, with four different types of neural networks used to support the data-driven part.
3) The proposed neural network for mapping the differences between measurements from the expanded model and

calculated values from the practical model to a reduced number of Koopman linear operators. This is an alternative to the neural network-based encoder–decoder structure explored in [24] and [25].

4) The calculation of Koopman modes from the expanded vector, containing direct measurements of state variables, algebraic variables, and indirect measurements.
5) The proposed optimization-based pattern reduction for a series of points in two axes (one representing initial conditions and other time samples), used for data compression and dynamic mode decomposition (DMD) [26].

The outline of the article is as follows: Section II introduces the problem formulation; in Section III, the TSA workflow with gray-box integration of the physics-driven (DAE-based) and data-driven (DNN-based) model is proposed; Section IV describes the reduction of a dynamic model by computational differential geometry, while Section V describes the compressed sensing, followed by the data-driven Koopman operator in Section VI; the proposed DNN-based architecture for the gray-box integration is given in Section VII; the proposed methodology is applied to a synchronous generator (SG) example in Section VIII; Section IX presents conclusions. The SG's input data are provided in the Appendix.

II. PROBLEM FORMULATION

The equation-driven framework for power system dynamics is based on nonlinear DAE [3]

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

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

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

The measurement vector is assumed to be of the form

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

In practice, $\boldsymbol{x}, \boldsymbol{z}$ are assumed to be known, while parameters $\boldsymbol{p}$ that fit the measurement vector $\boldsymbol{y}$ are unknown. If $\boldsymbol{f}, \boldsymbol{g}$ in (1), (2) are known, different methods are used for the calculation of $\boldsymbol{p}$ (typically optimization-based—see [16]–[18] and references therein).

We consider the setup in which system outputs ($\boldsymbol{y}$) are measured (or generated by an inaccessible model) or predicted by a higher-order DNN-based model operating on physics-derived state ($\boldsymbol{x}$) and algebraic variables ($\boldsymbol{z}$); these will be compared with values calculated by the practical (DAE-based) model, described by

$$\dot{\boldsymbol{x}}_p = \boldsymbol{f}_p\left(\boldsymbol{x}_p, \boldsymbol{z}_p, \boldsymbol{p}_p, t\right) \tag{4}$$

$$0 = \boldsymbol{g}_p\left(\boldsymbol{x}_p, \boldsymbol{z}_p, \boldsymbol{p}_p, t\right) \tag{5}$$

and the calculated (from practical model) measurement vector

$$\boldsymbol{y}_p = \boldsymbol{h}_p\left(\boldsymbol{x}_p, \boldsymbol{z}_p, \boldsymbol{p}_p, t\right) \tag{6}$$

with a known set of parameters ($\boldsymbol{p}_p$).

In this article, we develop a hybrid approach that retains physical state ($\boldsymbol{x}_p$) and algebraic variables ($\boldsymbol{z}_p$) from the DAE-based practical model. This portion is certified via information-geometric tools to be a robustly identifiable set of parameters ($\boldsymbol{p}_p$) from a given measurement structure ($\boldsymbol{y}$). However, some differences between actual measurements $\boldsymbol{y}$ and values calculated by (6) always exist, due to possible time variation in $\boldsymbol{p}_p$, unmodeled dynamics, different initial conditions, etc. In gray-box modeling, the refinements of $\boldsymbol{y}_p$ are driven by the difference between the two, as shown in (7). For an illustration of the proposed gray-box model, see the block diagram in Fig. 1

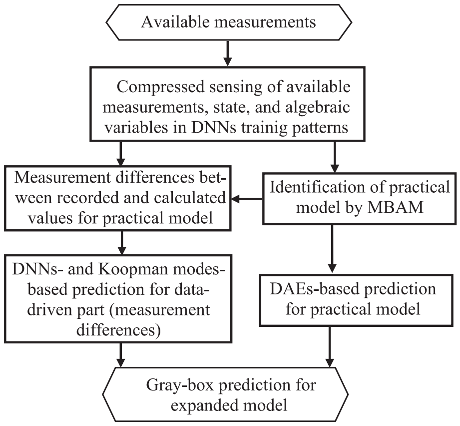


Fig. 1. Block diagram of the proposed gray-box integration of the physics-driven (DAE-based) and data-driven (DNN-based) models.

$$\Delta \boldsymbol{y} = \boldsymbol{y} - \boldsymbol{y}_p. \tag{7}$$

Historical database of the recorded transient in point of connection (PoC) of any local dynamic element can be vast. However, the similarity among these transients for small variations of initial condition is quite high. Hence, we first apply compressed sensing on training sets to improve the computation speed. Then, because physics-based (DAE-based) models can never fully replicate the recorded transients in PoC, irrespective of their complexity, we decide to form only a reasonable-sized practical DAE model, obtained using MBAM (right branch of Fig. 1). The idea is then to model the difference between recorded measurements in PoC and transients obtained by the practical model using the fully data-driven part. For this task, we use linear Koopman modes and DNN (left branch of block-diagram). Combining these two models (physics-based + data-driven) gives a gray-box expanded model. It should be noted that full transient responses of state, algebraic, and other boundary variables in PoC are calculated using only the initial conditions of recorded measurements in PoC of the local dynamic model.

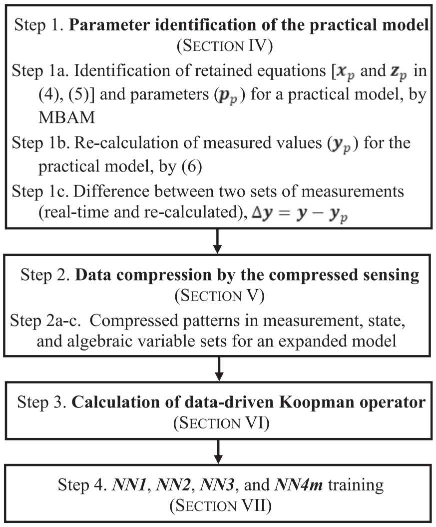
Based on the above analysis of our gray-box (hybrid) approach, we introduce the following definitions:

Definition 1: The practical model described by DAE (4)–(6) is physics-driven, and the (calculated) "measured" variables are $\boldsymbol{y}_p$ in (6), with a user-defined level of detail (with respect to the model complexity and the required set of parameters $\boldsymbol{p}_p$).

Definition 2: The expanded model is a union of the practical model with the data driven-part, trying to match a DAE (1)–(3), more specifically a set of "measured" data $\boldsymbol{y}$ in (3).

Note here that all measurements in the PoC of the analyzed local model may be divided into the following three groups (for more information, see Section VIII).

1) Direct measurements of state variables, $\boldsymbol{y}_{\boldsymbol{x}} \in \boldsymbol{x}$ (for example, measurement of rotor speed).
2) Direct measurements of algebraic variables, $\boldsymbol{y}_{\boldsymbol{z}} \in \boldsymbol{z}$ (for example, measurement of bus voltage magnitude). All algebraic variables related to the bus of interest are measured.
3) Indirect measurements of mixed nonlinear functions of the state ($\boldsymbol{x}$) and algebraic variables ($\boldsymbol{z}$), $\boldsymbol{y}_{\boldsymbol{x}\boldsymbol{z}}$ (for example, measurement of real power generation).



(a)

Fig. 2. Algorithms of training and verification phases of gray-box DAEs- and DNN-based TSA. (a) Training phase. (b) Verification phase (dynamic prediction model-based TSA)—for data flow, see Fig. 3.

III. WORKFLOW WITH INTEGRATED GRAY-BOX PHYSICS-DRIVEN (DAE-BASED) AND DATA-DRIVEN (DNN-BASED) MODELS FOR TSA

The workflow of the proposed gray-box environment with three main components (MBAM-based dynamic model and parameter reduction—Section IV, compressed sensing—Section V, and the Koopman theory-based dynamics prediction—Section VI) is shown in Fig. 2, while the data flow for the verification phase is shown in Fig. 3. Note that in Fig. 3 (and later in Fig. 4) $\oplus$ denotes the union of the data vectors, and $\otimes$ denotes the extension/reduction of the data vectors.

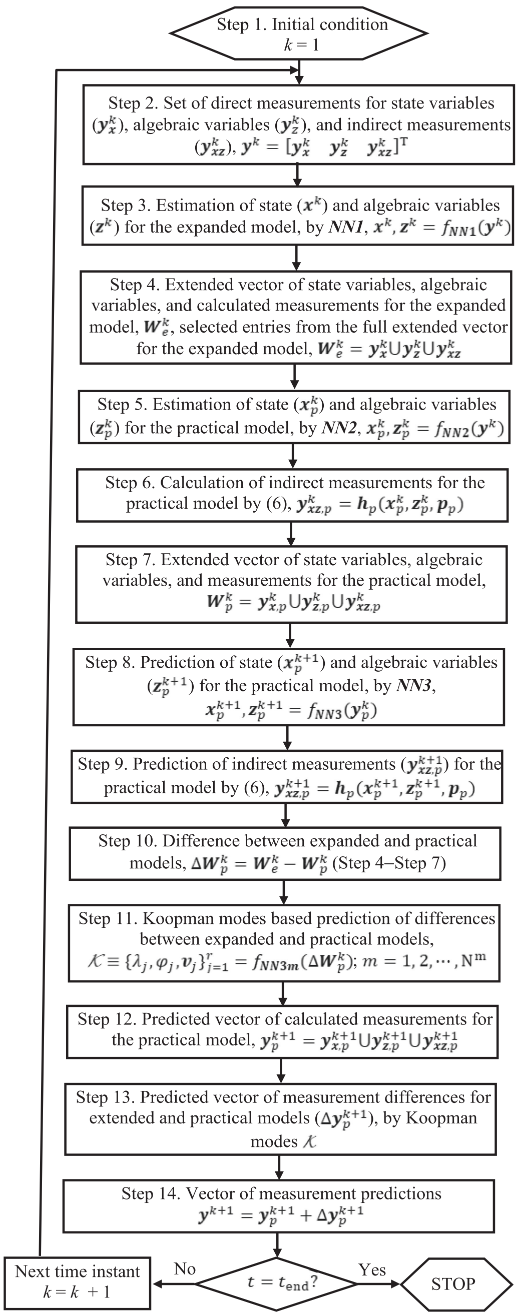
For the workflow in Fig. 2(b), it is interesting to note that for the prediction of state ($\boldsymbol{x}_p^{k+1}$) and algebraic variables ($\boldsymbol{z}_p^{k+1}$) in the practical model (in Step 8), the local DAE-based integration cannot be applied, because the power system balance equations in the PoC are unknown for the $(k+1)$st time instance without integrating the DAE for the whole power system.

IV. PARAMETER IDENTIFICATION OF THE PRACTICAL MODEL

Physics-based models complement purely data-driven approaches by incorporating *a priori* information that extends predictive performance beyond the training set. However, typical models suffer from the proliferation of unknown parameters if trained directly on raw data sets. These data sets are large, but also compressible, so the actual information content is often much smaller. This makes raw data sets unsuitable for learning physical parameters directly. We first perform the parameter identifiability and reduction step to produce a practical gray-box model that retains the relevant causal relationships. The states of this practical model are then retained in data-driven DNN model.

Our approach is based on information geometry and gives a global identifiability analysis, described in [3] and [23]. Briefly, a multiparameter model is interpreted as a Riemannian manifold on which parameter identifiability induces a distance metric. In this metric, the manifold is bounded, and the unidentifiable parameters correspond to narrow widths. The reduction is accomplished through the MBAM which approximates the complete model manifold by its boundary. Boundaries correspond to physically interpretable approximations, such as singular perturbation, but are identified in a data-driven way by calculating geodesics (i.e., distance-minimizing curves) on the model manifold. The reduced models are similarly expressed as a set of DAE and retain their physical interpretation as combinations of both state variables and parameters. MBAM is performed iteratively so that the number of parameters in the reduced model is commensurate with the information content of the data.

Note that the MBAM algorithm is not a contribution of this article. However, it is an essential step in our overall algorithm.



(b)

Fig. 2. (Continued.)

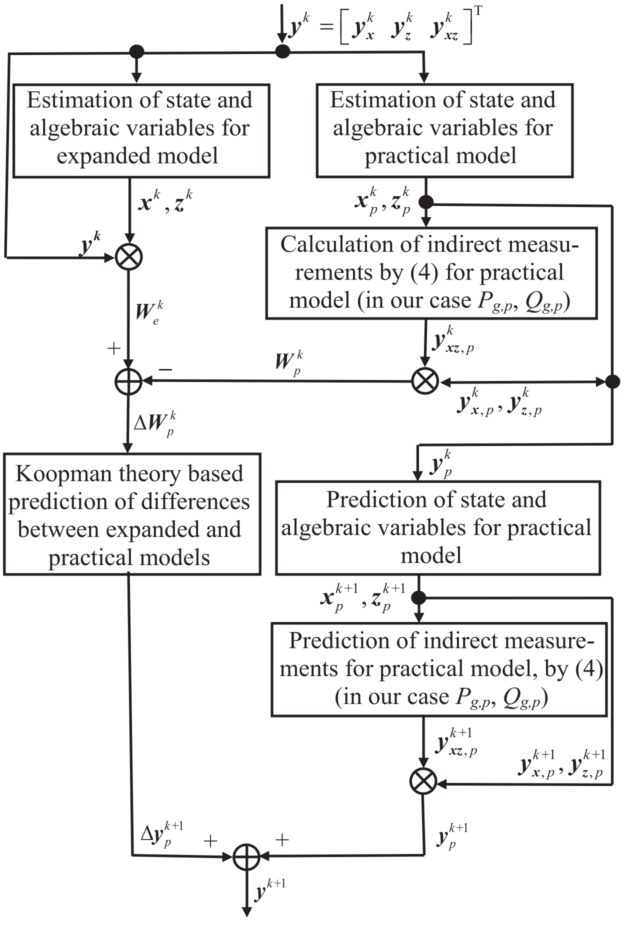


Fig. 3. Data flow of the proposed gray-box and DNN-based dynamic prediction model for TSA.

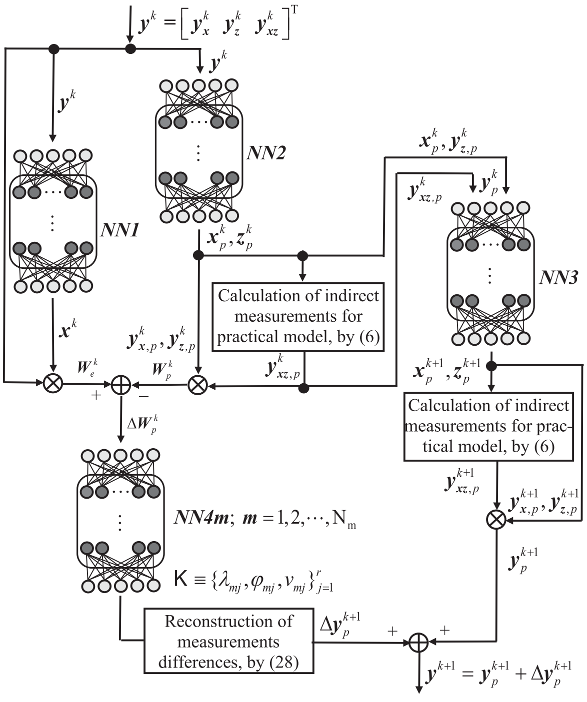


Fig. 4. DNN-based architecture for TSA in integrated gray-box physics-driven (DAE-based) and data-driven (DNN-based) dynamic models.

It selects a minimal physics-based model with a reasonable number of equations and parameters to cover the dominant dynamics. (For example, in the case of the analyzed SG, the dominant state variables are rotor angle and speed, as well as rotor voltage in *d*-axis.) Errors generated by using this simplified (practical) model in the proposed algorithm (7) are compensated by the data-driven part. This means that the proposed algorithm is not an equivalencing method because the practical model is only the internal step in the overall gray-box algorithm (see Figs. 1–3).

V. COMPRESSED SENSING

The data sets in power systems can be quite heterogeneous, as they come with a wide range of time stamps (from several seconds for SCADA measurements to 50–100 ms for quantities calculated by TSA software, to submilliseconds for raw PMU measurements). For the TSA task, higher granularity is needed compared to some other power system problems. Consequently, for different initial conditions followed by the daily load/generation variations, the number of measurement points can be huge. However, the similarity (correlation) of the data in two axes (time and initial condition) is high. This suggests that data compression is possible (see results in Section VIII), reducing the necessary storage capacity and, more importantly regarding the computation speed, the training set for neural networks. In this section, a well-known compressed sensing methodology [27] is applied for data compression of power system dynamics data with respect to time and initial condition axes.

The idea behind data compression is that there exists a new set of coordinates (transformed basis) to which a given initial set of data can be mapped and the new (transformed) data will be sparse. This means that a compressible signal $\boldsymbol{y} \in \mathbb{R}^{\mathrm{N_t}}$ may be written as a sparse vector $\boldsymbol{s} \in \mathbb{R}^{\mathrm{N_t}}$ on a transform basis $\boldsymbol{\Psi} \in \mathbb{R}^{\mathrm{N_t} \times \mathrm{N_t}}$ [2], [27], [28], or $\boldsymbol{y} = \boldsymbol{\Psi s}$.

Thus, after compression, only the vector corresponding to the sparse signal ($\boldsymbol{s}$) will need to be stored, rather than the entire measurement vector $\boldsymbol{y}$. If, for example, the $\mathrm{N_t}$-dimensional signal $\boldsymbol{y}$ is *K*-sparse in $\boldsymbol{\Psi}$, then only compressed signal $\boldsymbol{y}^{\mathrm{com}} \in \mathbb{R}^p$ with $K < p \ll \mathrm{N_t}$ needs to be stored as (similar compression can be applied for state variables vector $\boldsymbol{x}$)

$$\boldsymbol{y}^{\mathrm{com}} = \boldsymbol{\Phi y} = \boldsymbol{\Phi \Psi s} \tag{8}$$

where the measurement matrix $\boldsymbol{\Phi} \in \mathbb{R}^{p \times \mathrm{N_t}}$ represents a set of p linear measurements on the signal vector ($\boldsymbol{y}$).

The sparsest solution of (8) ($\hat{\boldsymbol{s}}$) satisfies the following optimization problem:

$$\hat{\boldsymbol{s}} = \underset{\boldsymbol{s}}{\operatorname{argmin}} \sum_{k=1}^{\mathrm{N_t}} |s_k| \tag{9}$$

subject to

$$\boldsymbol{y}^{\mathrm{com}} = \boldsymbol{\Phi \Psi s}. \tag{10}$$

Based on the assumptions about the low dimensionality of the (unknown) manifold underlying the system behavior, our expectation is that accurate, detailed transient responses can be recovered from relatively few discrete time points and initial conditions. In [23], it is shown that there are two ways for compression of recorded measurements.

1) Reduction of sampling points in both the time and initial condition axes (for suitably slowly varying signals).

2) Reduction in the number of elements in a transform basis (vector s), or a reduction in the number of needed fast Fourier transformation coefficients.

Note that the optimization in (9), representing compressed sensing in the strict sense, corresponds to the latter. However, this approach is not suitable for the proposed algorithm because it produces missing inputs/outputs for DNN training (see Section VII). Therefore, we implement a modification (the former approach) in the following way.

All inputs (measurements) may be arranged into a 3D data tensor as $\boldsymbol{Y}(m, w, t)$ [the state tensor $\boldsymbol{X}(m, w, t)$ can be reduced similarly]. Formally, let $\mathcal{M}$ denote an ensemble of $\mathrm{N_m}$ sets of measurements ($\mathrm{N_m}$ is the number of measurements) and $\mathcal{W}$ denote an ensemble of $\mathrm{N_w}$ sets of initial conditions ($\mathrm{N_w}$ is the number of initial conditions). For each set of measurements, $m \in \mathcal{M}$, and set of initial conditions, $w \in \mathcal{W}$, we observe a trajectory $\boldsymbol{y}(m, w, t)$ of length N_t of the system variables, where $t = 1, 2, \ldots, \mathrm{N}_t$ denotes the time samples. The input (measurement) data tensor has dimensions $(\mathrm{N_m} \times \mathrm{N_w} \times \mathrm{N_t})$. For the mth measurement type, we get

$$\boldsymbol{Y}_m = \begin{bmatrix} -\boldsymbol{y}_m(w_1) - \\ -\boldsymbol{y}_m(w_2) - \\ \vdots \\ -\boldsymbol{y}_m(w_{\mathrm{N_w}}) - \end{bmatrix} = \begin{bmatrix} | & | & & | \\ \boldsymbol{y}_m(t_1) & \boldsymbol{y}_m(t_2) & \cdots & \boldsymbol{y}_m(t_{\mathrm{N_t}}) \\ | & | & & | \end{bmatrix} = \begin{bmatrix} y_{m,11} & y_{m,12} & \cdots & y_{m,1,\mathrm{N_t}} \\ y_{m,21} & y_{m,22} & \cdots & y_{m,2,\mathrm{N_t}} \\ \vdots & \vdots & \ddots & \vdots \\ y_{m,\,\mathrm{N_w},1} & y_{m,\mathrm{N_w},2} & \cdots & y_{m,\mathrm{N_w},\mathrm{N_t}} \end{bmatrix} \quad (11)$$

where $y_{m,w,t}$; $m = 1, 2, \ldots, \mathrm{N_m}$; $w = 1, 2, \ldots, \mathrm{N_w}$; $t = 1, 2, \ldots, \mathrm{N_t}$ is an element of $\boldsymbol{y}_m(w)$ or $\boldsymbol{y}_m(t)$.

For the reduction of sampling points, two metrics for initial conditions ($w = 1, 2, \ldots, \mathrm{N_w}$) and time points ($t = 1, 2, \ldots, \mathrm{N_t}$) based on ℓ_2-distances are proposed, respectively

$$D\left[\boldsymbol{y}_{w_i}(m,t), \boldsymbol{y}_{w_j}(m,t)\right] = \sum_{m}^{\mathrm{N_m}} \sum_{t=1}^{\mathrm{N_t}} \left(y_{m,w_i,t} - y_{m,w_j,t}\right)^2$$

$$w_i = 1, 2, \ldots, \mathrm{N_w}; w_j = w_i + 1, w_i + 2, \ldots, \mathrm{N_w} \quad (12a)$$

$$D\left[\boldsymbol{y}_{t_i}(m,w), \boldsymbol{y}_{t_j}(m,w)\right] = \sum_{m}^{\mathrm{N_m}} \sum_{w=1}^{\mathrm{N_w}} \left(y_{m,w,t_i} - y_{m,w,t_j}\right)^2$$

$$t_i = 1, 2, \ldots, \mathrm{N_t}; t_j = t_i + 1, t_i + 2, \ldots, \mathrm{N_t}. \quad (12b)$$

These two metrics are used for the reduction of sampling series of measurements ($\boldsymbol{Y}$) and state variables ($\boldsymbol{X}$) for the practical model, as

$$D\left(\boldsymbol{Y}_{w_i}, \boldsymbol{Y}_{w_j}\right) + D\left(\boldsymbol{X}_{w_i}, \boldsymbol{X}_{w_j}\right) + D\left(\boldsymbol{Y}_{p,w_i}, \boldsymbol{Y}_{p,w_j}\right) + D\left(\boldsymbol{X}_{p,w_i}, \boldsymbol{X}_{p,w_j}\right) \le \varepsilon_w \quad (13)$$

$$D\left(\boldsymbol{Y}_{t_i}, \boldsymbol{Y}_{t_j}\right) + D\left(\boldsymbol{X}_{t_i}, \boldsymbol{X}_{t_j}\right) + D\left(\boldsymbol{Y}_{p,t_i}, \boldsymbol{Y}_{p,t_j}\right) + D\left(\boldsymbol{X}_{p,t_i}, \boldsymbol{X}_{p,t_j}\right) \le \varepsilon_t \quad (14)$$

where ε_w and ε_t are tuneable hyperparameters, representing thresholds for the reduction of the initial condition and time patterns, respectively, with terms in (13) and (14) defined as in (12).

TABLE I
REDUCTION OF TRAINING PATTERNS FOR DNN

	Reduced initial conditions	Reduced time points	Storage capacity [MB]			
			$\boldsymbol{y}$	$\boldsymbol{x}$	$\boldsymbol{z} = \boldsymbol{z}_p$	$\boldsymbol{x}_p$
Unreduced	$\mathrm{N_w} = 1440$	$\mathrm{N_t} = 400$	26.8	21.8	15.5	10.9
$\varepsilon_w = 0.02$;$\varepsilon_t = 0.02$	$\mathrm{N_w^{com}} = 1140$	$\mathrm{N_t^{com}} = 400$	19.4	15.8	11.2	7.9
$\varepsilon_w = 0.1$; $\varepsilon_t = 0.1$	$\mathrm{N_w^{com}} = 783$	$\mathrm{N_t^{com}} = 393$	15.1	12.2	8.7	6.1
$\varepsilon_w = 0.5$; $\varepsilon_t = 0.5$	$\mathrm{N_w^{com}} = 554$	$\mathrm{N_t^{com}} = 378$	10.4	8.4	6.0	4.2
$\varepsilon_w = 0.5$; $\varepsilon_t = 10$	$\mathrm{N_w^{com}} = 554$	$\mathrm{N_t^{com}} = 262$	7.2	5.9	4.2	2.9

As a consequence of this optimization, the number of initial conditions is reduced to $\mathrm{N_w^{com}}$ and the number of time points to $\mathrm{N_t^{com}}$ (to simplify the notation, superscript $^{\mathrm{com}}$ is omitted in the rest of the text, except in Table I, where it is needed for clarity).

Detection of bad data in transient responses (due to measurement errors) of training patterns $\boldsymbol{Y}$ is performed by a manifold learning-based data clustering [1]. Input for the verification phase (real-time TSA) of the proposed algorithm is only the initial condition of state/algebraic variables (see Section VIII). In this case, the bad data is identified by the static state estimation.

VI. DATA-DRIVEN KOOPMAN OPERATOR

The Koopman operator provides a global linear infinite-dimensional representation of a nonlinear dynamical system [2], [29], [30]. For the real-valued, multiple observation (input or measurement) functions $\mathbb{M} \to \mathbb{R}$ ($\mathbb{M}$ denotes the state space, and $\mathbb{R}$ denotes the scalar measurement space), which are elements of an infinite-dimensional Hilbert space, the input (measurement) model (1)–(3), formally replacing vector of algebraic variables ($\boldsymbol{z}$) into (1), (3), can be rewritten in compact form for $\mathrm{N_m}$ measurements as [note that in our case this approach is used for identification of data-driven differences between real-world measurements of the model (3) and values calculated by the practical model (6)] (see also Section VII)

$$\boldsymbol{y} = \boldsymbol{g}(\boldsymbol{x}) = \begin{bmatrix} g_1(\boldsymbol{x}) \\ g_2(\boldsymbol{x}) \\ \vdots \\ g_{\mathrm{N_m}}(\boldsymbol{x}) \end{bmatrix}. \quad (15)$$

The linear Koopman operator ($\mathcal{K}_t$) is an infinite-dimensional operator, acting on a vector-valued Hilbert space of measurement functions, $\boldsymbol{g}(\boldsymbol{x})$, as [2, eq, (7.51)], [29], [30]

$$\mathcal{K}_t \boldsymbol{g}(\boldsymbol{x}) \triangleq \boldsymbol{g}(\boldsymbol{x}) \circ \boldsymbol{F}_t \quad (16a)$$

where $\boldsymbol{F}_t$ is a transition function of states ($\boldsymbol{x}$) in state-space ($\mathbb{M}$), determined by a set of DAE (1)–(3), or (4)–(6) and o is the composition operator.

The linear characteristics of the Koopman operator ($\mathcal{K}_t$) allow us to perform eigendecomposition of $\mathcal{K}_t$ as

$$\mathcal{K}_t \varphi_j(\boldsymbol{x}) = \lambda_j \varphi_j(\boldsymbol{x}); j = 1, 2, \ldots, \infty \quad (16b)$$

where eigenvalues and eigenfunctions of $\mathcal{K}_t$ are λ_j and $\varphi_j(\boldsymbol{x})$, respectively. The eigenfunctions $\boldsymbol{\varphi}_j(\boldsymbol{x})$ are the inner product of the state vector ($\boldsymbol{x}$) with the left eigenvector of the linear Koopman operator [2].

For the discrete system with sufficiently small time step (Δt) for kth time step we can write, akin to (16a)

$$\mathcal{K}_{\Delta t} \boldsymbol{g}\left(\boldsymbol{x}^k\right) = \boldsymbol{g}\left(\boldsymbol{x}^k\right) \circ \boldsymbol{F}_{\Delta t} = \boldsymbol{g}\left(\boldsymbol{x}^{k+1}\right) \quad (17)$$

where $\boldsymbol{x}^k = \boldsymbol{x}(t_k)$, $\boldsymbol{x}^{k+1} = \boldsymbol{x}(t_{k+1})$, $t_{k+1} = t_k + \Delta t$. This means that the Koopman operator defines an infinite-dimensional linear mapping as the function of inputs (measurements) to the next time step [see [2], Fig. 7.10].

Individual mth input (measurement) may be expanded in terms of eigenfunctions $\varphi_j(\boldsymbol{x})$, similar to eigendecomposition in (16b), providing the basis for a Hilbert space

$$y_m = g_m\left(\boldsymbol{x}^k\right) = \sum_{j=1}^{\infty} \varphi_{mj}\left(\boldsymbol{x}^k\right)\boldsymbol{v}_{mj} \tag{18}$$

where $\boldsymbol{v}_{mj}$ is the jth Koopman mode associated with eigenfunction $\varphi_{mj}(\boldsymbol{x}^k)$ of the mth input (measurement).

Based on the transition in (17), it is possible to represent the dynamics of the mth input (measurement) [$g_m(\boldsymbol{x})$] as [2, eq. (7.69)]

$$\begin{aligned} y_m^{k+1} &= g_m\left(\boldsymbol{x}^{k+1}\right) = \mathcal{K}^k_{m\Delta t} g_m\left(\boldsymbol{x}^0\right) \\ &= \mathcal{K}^k_{m\Delta t} \sum_{j\,=\,1}^{\infty} \varphi_{mj}\left(\boldsymbol{x}^0\right)\boldsymbol{v}_{mj} \\ &= \sum_{j=1}^{\infty} \mathcal{K}^k_{m\Delta t}\varphi_{mj}\left(\boldsymbol{x}^0\right)\boldsymbol{v}_j = \sum_{j=1}^{\infty} \lambda^k_{mj}\varphi_{mj}\left(\boldsymbol{x}^0\right)\boldsymbol{v}_{mj} \end{aligned} \tag{19}$$

where $\boldsymbol{x}^0 = \boldsymbol{x}(t_0) = \boldsymbol{x}(t = 0)$ denotes the initial condition.

The sequence of triples, $\{\lambda_{mj}, \varphi_{mj}, \boldsymbol{v}_{mj}\}_{j\,=\,1}^{\infty}$; $m = 1, 2, \ldots \mathrm{N_m}$ is the Koopman mode decomposition for set of $\mathrm{N_m}$ inputs (measurements) [2], [29], [30].

DMD approximates the Koopman operator with a best-fit (finite) linear model from time-dependent measurements. There are several algorithms for identifying Koopman embeddings and eigenfunctions from data, such as extended mode decomposition [31], or QR decomposition of the input snapshot matrix [32]. For the method proposed in this article, the basic DMD and extended DMD (eDMD) algorithms are of primary interest, and these algorithms are described in more detail below.

A. Basic DMD Algorithm

The DMD is a data-driven algorithm, and the initial step is to collect a few pairs of state variable snapshots $\{\boldsymbol{x}^k, \boldsymbol{x}^{k+1}\}_{k\,=\,0}^{m-1}$ arranged into two data matrices [33]

$$\boldsymbol{X} = \begin{bmatrix} | & | & & | & \\ x(t_0) & x(t_1) & \cdots & x(t_{m-1}) & \cdots \\ | & | & & | & \end{bmatrix} \tag{20}$$

$$\boldsymbol{X'} = \begin{bmatrix} | & | & & | & \\ x(t_1) & x(t_2) & \cdots & x(t_m) & \cdots \\ | & | & & | & \end{bmatrix} \tag{21}$$

where

$$\boldsymbol{x}^{k+1} = \boldsymbol{A}\boldsymbol{x}^k. \tag{22}$$

A best-fit linear operator ($\boldsymbol{A}$) that maps $\boldsymbol{X}$ to $\boldsymbol{X'}$ is obtained as

$$\boldsymbol{A} = \underset{\boldsymbol{A}}{\operatorname{argmin}} \left\|\boldsymbol{X'} - \boldsymbol{A}\boldsymbol{X}\right\|_2^2 = \boldsymbol{X'} - \boldsymbol{X}^{\dagger} \tag{23}$$

where $\boldsymbol{X}^{\dagger}$ is the Moore–Penrose pseudoinverse of $\boldsymbol{X}$. It essentially tries to model transition $\boldsymbol{x}^k \rightarrow \boldsymbol{x}^{k+1}$. A good description of the algorithm is in [2, p. 240].

The DMD modes are eigenvectors of full matrix $\boldsymbol{A}$ corresponding to the eigenvalues in $\boldsymbol{\Lambda}$, $\boldsymbol{A\Phi} = \boldsymbol{\Phi\Lambda}$ (for derivation, see [30]). One of the most important aspects of the DMD is the ability to predict the system state in terms of a data-driven spectral decomposition

$$\boldsymbol{x}(t) = \sum_{j=1}^{r} v_j\left(\boldsymbol{x}^0\right) e^{\omega_j t} b_j = \boldsymbol{\Phi} e^{\boldsymbol{\Omega} t}\boldsymbol{b} \tag{24}$$

where r is the rank of the reduced model (depending on the selected truncation), φ_j (element of $\boldsymbol{\Phi}$) is DMD mode (eigenvector of the $\boldsymbol{A}$ matrix), the initial snapshot is $\boldsymbol{x}^0 = \boldsymbol{x}(t_0) = \boldsymbol{x}(t = 0)$, ω_j (element of $\boldsymbol{\Omega}$) is the imaginary part of DMD eigenvalue (λ_j), and b_j (element of $\boldsymbol{b}$) is the initial amplitude of each mode, calculated as

$$\boldsymbol{b} = \boldsymbol{\Phi}^{\dagger}\boldsymbol{x}^0. \tag{25}$$

B. eDMD Algorithm

The basic DMD algorithm uses the system's linear inputs (measurements) (as a function of state variables), which cannot fully characterize nonlinear phenomena. In [34], measurement vector (composed only of measurements of the states) is expanded to include indirect nonlinear measurements (eDMD), enriching the basis used to represent the Koopman operator.

In our eDMD case, the extended vector of state variables ($\boldsymbol{x}$), algebraic variables ($\boldsymbol{z}$), and measurements ($\boldsymbol{y}$) for the practical model from (15) is constructed as

$$\boldsymbol{W}_p = \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{z} \\ \boldsymbol{y} \end{bmatrix} \tag{26a}$$

where

$$\boldsymbol{x} = \left[x_1\ x_2\ \ldots\ x_{\mathrm{N}_{\boldsymbol{x}}}\right]^{\mathrm{T}} \tag{26b}$$

$$\boldsymbol{z} = \left[z_1\ z_2\ \ldots\ z_{\mathrm{N}_{\boldsymbol{z}}}\right]^{\mathrm{T}} \tag{26c}$$

$$\boldsymbol{y} = \left[g_1(\boldsymbol{x}, \boldsymbol{z})\ g_2(\boldsymbol{x}, \boldsymbol{z})\ \ldots\ g_{\mathrm{N_m}}(\boldsymbol{x}, \boldsymbol{z})\right]^{\mathrm{T}} \tag{26d}$$

and $\mathrm{N}_{\boldsymbol{x}}$, $\mathrm{N}_{\boldsymbol{z}}$, and $\mathrm{N_m}$ are the total numbers of state variables, algebraic variables, and indirect measurements, respectively.

Note here that all inputs in the extended vector of practical model ($\boldsymbol{W}_p$) are selected from sets of available direct measurements of state and algebraic variables, as well as indirect measurements (see Section II). In the case of redundant variables [for example, the rotor speed in the state ($\boldsymbol{x}$) and input (measurement) ($\boldsymbol{y}$) vectors], the union of nonredundant vectors $\boldsymbol{W}_p = \boldsymbol{x} \cup \boldsymbol{z} \cup \boldsymbol{y}$ is applied.

Two data matrices ($\Delta\boldsymbol{W}_p$ and $\Delta\boldsymbol{W}'_p$—see the workflow in Figs. 2 and 3) are constructed as in the basic DMD algorithm (23). A best-fit linear operator ($\boldsymbol{A}_{\Delta\boldsymbol{W}_p}$) that maps $\Delta\boldsymbol{W}_p$–$\Delta\boldsymbol{W}'_p$ [similar to (23)] is constructed as

$$\begin{aligned} \boldsymbol{A}_{\Delta\boldsymbol{W}_p} &= \underset{\boldsymbol{A}_{\Delta\boldsymbol{W}_p}}{\operatorname{argmin}} \left\|\Delta\boldsymbol{W'}_p - \boldsymbol{A}_{\Delta\boldsymbol{W}_p}\Delta\boldsymbol{W}_p\right\|_2^2 \\ &= \Delta\boldsymbol{W'}_p\Delta\boldsymbol{W}_p^{\dagger}. \end{aligned} \tag{27}$$

VII. Proposed DNN-Based Architecture for the Gray-Box Integration of Physics-Driven (DAE-Based) and Data-Driven Models

Workflow for the training and verification phases of TSA (see Fig. 2) and data flow (see Fig. 3) fully determine the proposed DNN-based architecture of gray-box integration of physics-driven (DAE-based) and data-driven (DNN-based) models.

DAE-based calculations are performed in cases where physics-driven descriptions are available. In cases where such descriptions are not available, only the data-driven (DNN-based) approach is applied.

Based on Fig. 4, the neural networks perform four types of calculations.

1) NN1: Mapping of measurements to calculated state variables for the expanded model, with $\mathrm{N_m}$ inputs [dimension of vector of measurements ($\boldsymbol{y}^k$), in (3)] and $\mathrm{N}_{\boldsymbol{x}}$ outputs [dimension of vector of state variables ($\boldsymbol{x}^k$), in (1)–(3)].
2) NN2: Mapping of measurements to calculated state variables for the practical model, with $\mathrm{N_m}$ inputs [dimension of vector of measurements ($\boldsymbol{y}^k$), in (3)] and $\mathrm{N}_{\boldsymbol{x}_p} + \mathrm{N}_{\boldsymbol{z}_p}$ outputs [dimensions of vector of calculated state and algebraic variables, $\boldsymbol{x}_p^k$ and $\boldsymbol{z}_p^k$ in (4)–(6), respectively].
3) NN3: Prediction of state and algebraic variables for the practical model, with $\mathrm{N}_{\boldsymbol{x}_p} + \mathrm{N}_{\boldsymbol{z}_p} + \mathrm{N}_{\boldsymbol{xz}_p}$ inputs [dimensions of the vector of states ($\boldsymbol{x}_p^k$), algebraic variables ($\boldsymbol{z}_p^k$), and mixed equality constraints ($\boldsymbol{y}_{\boldsymbol{xz},p}^k$), in (4)–(6)] and $\mathrm{N}_{\boldsymbol{x}_p} + \mathrm{N}_{\boldsymbol{z}_p}$ outputs.
4) NN4m; $m = 1, 2, \ldots, \mathrm{N_m}$: Mapping of measurement differences to Koopman linear operator, $\mathcal{K} \equiv \{\lambda_{mj}, \varphi_{mj}, v_j\}_{j\,=\,1}^{r}$, with N_t inputs [dimension of compressed vector of measurement differences for the practical model, $\Delta \boldsymbol{W}_p$] and $\mathrm{N}_{\mathcal{K}}$ outputs [number of output elements in the Koopman operator (24)].

Note that in DNN training phase, all vectors in Fig. 4 are replaced with tensors ($\boldsymbol{x} \rightarrow \boldsymbol{X}$ and $\boldsymbol{y} \rightarrow \boldsymbol{Y}$, see Section V).

The reconstruction of the future measurement differences (predictions of state and algebraic variable differences are not necessary for the rest of the algorithm) in the practical model (for example, for the mth measurement, Δy_{mp}^{k+1}) may be constructed for all future time instants using the Koopman linear operator (see Fig. 4). For example, by first rewriting as $\omega_{mj} = \ln(\lambda_{mj}) / \Delta t$, predictions of the measurement differences from (22) are given by [34]

$$\Delta y_{mp}^{k+1} = \sum_{j=1}^{r} v_{mj}\left(\boldsymbol{x}^0\right) e^{\omega_{mj} t_{k+1}} b_{mj}; m = 1, 2, \ldots, \mathrm{N_m} \tag{28}$$

where

r - rank of the reduced model in (24), depending on the selected truncation;

$v_{mj}(\boldsymbol{x}^0)$ - eigenvector of jth mode;

b_{mj} - initial amplitude of jth mode (25);

ω_{mj} - frequency of jth mode.

The optimal truncation (optimized hard threshold) for eDMD (see Section VIII-B) can be performed by following [35].

VIII. APPLICATION

The proposed integration of data- and physics-driven power system dynamic model was tested on a real-world test system with 441 buses, 655 branches (lines and transformers), 67 SGs (37 with fourth-order dynamic models and 30 with sixth-order dynamic models), equipped with automatic voltage regulators and turbine-governor dynamic models. The dynamic model has 797 and 1284 differential (state) and algebraic variables, respectively—all system data are available in [36].

A. Input Data

Input (measurement) set [$\boldsymbol{y}(t)$ in (3) and $\boldsymbol{Y}(m, w, t)$ in (11)] is obtained by TSA of the sixth-order state and fourth-order algebraic SG's dynamic models in Matlab-based PSAT software [37], with input data for analyzed SG provided in Appendix.

The rest of the proposed algorithm (the compressed sensing, the Koopman linear operators, and the gray-box integration of DAE- and DNN-based models, shown in Fig. 4) is implemented in the Matlab environment, while the information geometry-based dynamic model reduction is in Julia.

Transient responses are generated by applying the three-phase short circuit on the connection bus, cleared after 250 ms. Transients after fault clearing were recorded for 20 s, with the time step of 0.05 s. The total of 1440 (every 60 s, or 60 × 24) initial conditions for forecasted daily generation curves in the whole test system (for thermal and hydro units) in conjunction with 11 forecasted daily load curves (for residential, commercial, different industry load types, etc.) are simulated by full (used as a source of measured values) and reduced models.

We have recorded the following measurements [$\boldsymbol{y}$ in (3)]:

1) state variable (from vector $\boldsymbol{x}$) for rotor speed, $\omega(t)$;
2) algebraic variables (from vector $\boldsymbol{z}$) for nodal voltage magnitude, $V(t)$ and angle, $\theta(t)$;
3) algebraic variable (from vector $\boldsymbol{z}$) for exciter's output voltage, $v_f(t)$ (representing the exciter's dynamics);
4) algebraic variable (from vector $\boldsymbol{z}$) for turbine governor's output mechanical power, $P_m(t)$ (representing the turbine governor's dynamics); and
5) mixed nonlinear functions [from $\boldsymbol{g}$ in (2)] of state ($\boldsymbol{x}$) and algebraic variables ($\boldsymbol{z}$) for active, $P_g(t)$ and reactive, $Q_g(t)$ power generations, or

$$\begin{aligned}
\boldsymbol{x} &= \left[\delta\ \omega\ \ e_q^{'}\ e_d^{'}\ e_q^{''}\ e_d^{''}\ \right]^{\mathrm{T}};\\
\boldsymbol{z} &= \left[V\ \ \theta\ v_f\ P_m\ \right]^{\mathrm{T}};\\
\boldsymbol{g} &= \ \{P_g; Q_g\}\,;\\
\boldsymbol{y} &= \left[\omega\ V\ \ \theta\ v_f\ \ P_m\ \ P_g\ Q_g\ \ \right]^{\mathrm{T}};\\
\boldsymbol{y}_{\boldsymbol{x}} &= \omega; \boldsymbol{y}_{\boldsymbol{z}} = \left[V\ \theta\ \ v_f\ P_m\ \right]^{\mathrm{T}}; \boldsymbol{y}_{\boldsymbol{xz}} = [\ P_g\ Q_g]^{\mathrm{T}};\\
\boldsymbol{W} &= \boldsymbol{x} \cup \boldsymbol{z} \cup \boldsymbol{y}\\
&= \left[\delta\ \ \omega\ \ e_q^{'}\ e_d^{'}\ e_q^{''}\ e_d^{''}\ \ \ V\ \theta\ \ v_f\ \ \ P_g\ Q_g\ \ \right]^{\mathrm{T}}.
\end{aligned}$$

where measurements $V(t)$, $\theta(t)$, $P_g(t)$, and $Q_g(t)$ represent the interaction of dynamics of local SG with the rest of the power system in the connection point.

To simulate the real-world measurements, random noise of ±2 % is added to the calculated values in measurement vector $\boldsymbol{y}$ (used to prepare the training patterns $\boldsymbol{Y}$ and $\boldsymbol{X}$ for neural networks), with the following variances:

1) 10^{-2} for the SG's speed (ω), voltage magnitude (V), exciter's output voltage (v_f), and turbine governor's output mechanical power (P_m);
2) 10^{-3} for voltage angle (θ) measurements; and
3) 10^{-1} for indirect active (P_g) and reactive power generation (Q_g) measurements.

For training of ***NN1*** and ***NN2***, the unmeasured state variables [state vector $\boldsymbol{x}$, or tensor $\boldsymbol{X}$] in a sixth-order model are following:

1) state variable for rotor angle, $\delta(t)$, calculated as in [38]; note that transient response of this state variable is also used for the vector of state variables in the practical model ($\boldsymbol{x}_p$);
2) state variables for transient rotor voltages in *q*-, $e'_q(t)$, and *d*-axis, $e'_d\,(t)$, calculated by the methodology from [39]; and
3) state variables for subtransient rotor voltages in *q*-, $e''_q(t)$, and *d*-axis, $e''_d(t)$, calculated by the methodology from [39].

B. Practical Model

The DAE-based SG's dynamic model is reduced by the MBAM algorithm to (as explained in Section IV and [3]):

1) three differential equations [$\boldsymbol{f}_p$ in (4)] for δ, ω, and e'_d;
2) one differential equation transformed to the algebraic equation for e'_q ($T'_{d0} \to \infty$);
3) two retained algebraic equations [$\boldsymbol{g}_p$ in (5)] for P_g and Q_g, or

$$\begin{aligned}
\boldsymbol{x}_p &= \begin{bmatrix} \delta \; \omega \; e'_d \end{bmatrix}^{\mathrm{T}} \\
\boldsymbol{z}_p &= \boldsymbol{z} = \begin{bmatrix} V \;\; \theta \; v_f \;\; P_m \end{bmatrix}^{\mathrm{T}} \\
\boldsymbol{g}_p &= \boldsymbol{g} = \{P_g; Q_g\} \\
\boldsymbol{y}_{\boldsymbol{x},p} &= \omega; \boldsymbol{y}_{\boldsymbol{z},p} = \begin{bmatrix} V \;\theta \;\; v_f \;\; P_m \end{bmatrix}^{\mathrm{T}}; \boldsymbol{y}_{\boldsymbol{xz},p} = [\; P_g \; Q_g]^{\mathrm{T}} \\
\boldsymbol{W}_p &= \boldsymbol{x}_p \cup \boldsymbol{z}_p \cup \boldsymbol{y}_p \\
&= \begin{bmatrix} \delta \;\; \omega \; e'_d \;\; V \; \theta \;\; v_f \; P_m \;\;\; P_g \; Q_g \end{bmatrix}^{\mathrm{T}}.
\end{aligned}$$

Note that the algebraic variable e'_q is dependent and may be calculated directly from $\boldsymbol{x}_p$ and $\boldsymbol{z}_p$ as [37]

$$e'_q = \frac{x'_d}{x_d} v_f + \frac{\left(x_d - x'_d\right)}{x_d} V \cos\left(\delta - \theta\right). \tag{29}$$

Also the extended vector for the expanded system, $\boldsymbol{W}_e$ in Figs. 3 and 4 is with the same elements as in $\boldsymbol{W}_p$ (selected from $\boldsymbol{W} = \boldsymbol{x} \cup \boldsymbol{z} \cup \boldsymbol{y}$).

C. Reduction of Training Tensors

Initial conditions vary slowly with daily generation/load profiles, so the transients generate a family of similar curves. Transient responses for training patterns of state, algebraic, and measurement variables for expanded model ($\boldsymbol{W}_e$ in Figs. 3 and 4) are shown in Fig. 5 (due to limited space, we show only some representative traces for ω, e'_q, V, P_m, P_g, and Q_g).

The reduction of training patterns is performed using tensor's deviations (13) and (14) for initial conditions and time samples, respectively. The obtained results are presented in Table I. From these results, we note that the storage capacity is reduced by >60 %. However, note that there are still some redundant sets (for example, ω in $\boldsymbol{x}$ and $\boldsymbol{y}$, V in $\boldsymbol{z}$ and $\boldsymbol{y}$, etc.), with additional opportunity for savings of the storage capacity.

Transient responses of reduced training patterns of state, algebraic, and measurement variables for the practical model ($\boldsymbol{W}_p$ in Figs. 3 and 4) are shown in Fig. 6 (matching the traces from Fig. 5). These results are obtained with the reduced Koopman model rank $r = 0.2$ (28). This low truncation rank suffices to capture the data-driven dynamics, which is confirmed in our power system simulations [1], and in references from different engineering areas, such as [2], [21], [22]; for a derivation of the optimal hard threshold see [35].

Four-layer neural networks (input layer, two hidden layers, and output layer) in DNN-based architecture shown in Fig. 4, used to identify the Koopman mode decomposition and dynamic predictions in gray-box hybrid modeled power system, are trained with reduced training patterns from the last row in Table I. Please note that these data are related only to the local dynamic model (one SG in our case) and measurements in PoC to the main part of the power system. Basic inputs for the training of neural networks in Fig. 4 are presented in Table II. The numbers of neurons for input and output layers are determined by the number of measurements, the dimension of the practical model, and the selected truncation rate for ***NN4m***. Only the number of neurons in hidden layers is chosen, taking into account the size of the training pattern (because of limited space, the sensitivity of the results to neural networks architecture is omitted from the article).

For the verification phase of DNN-based architecture in Fig. 4, the initial condition (for $t = 0$) of measurement set ($\boldsymbol{y}$) (different from training patterns) is shown in (30), while the remaining elements in the initial state vector ($\boldsymbol{x}^0$) are obtained from $\boldsymbol{y}(t = 0)$ using the differential equations of the practical model (Section VIII-B)

$$\begin{aligned}
\boldsymbol{y}\,(t = 0) &= \begin{bmatrix} \omega_0 \; V_0 \;\; \theta_0 \; v_{f0} \;\; P_{m0} \;\;\; P_{g0} \; Q_{g0} \end{bmatrix}^{\mathrm{T}} \\
&= [1.0046 \;\; 0.8473 \;\; 0.5544 \;\; 4.6184 \;\; 0.1459 \\
&\qquad 13.2740 \;\; -1.4515]^{\mathrm{T}}.
\end{aligned} \tag{30}$$

For this initial condition, predicted transient responses are shown in Fig. 7. The obtained mean squared errors (MSEs) for trained neural networks ($\cdot 10^{-6}$) approximately are: ***NN1*** (231), ***NN2*** (13200), ***NN3*** (128), and ***NN4m*** (247, 8760, 14000, 580, 1440, 2220, and 2020, respectively). Calculation time for obtaining these 20 s time responses is approximately 2.4 s, which is lower than ~17 s for full DAE-based transient analysis (recorded on the personal computer with the following performances: Intel(R) Core(TM) i7-6860HQCPU @ 2.70 GHz, 64-bit Operating System, 32 GB RAM) and comparable with other state of the art transient algorithms [40, Table II].

From Fig. 7 we can conclude that the differences in transient responses between expanded and practical models for some quantities may be increased [for example, for $Q_g(t)$]. For these cases, the MSEs of trained neural networks determine the predicted responses due to the superposition of accumulated errors at the outputs of all neural networks along the time axis. Neural networks ***NN2***, ***NN3***, and ***NN4m*** have a dominant influence on these errors.

From the results presented in Fig. 7 we can conclude that: 1) practical (reduced) model alone may produce large differences in transient responses of some variables (particularly for voltage magnitude, voltage angle, and SG's reactive power), and 2) the data-driven and DNN-based correction (in the proposed overall gray-box hybrid environment) is very useful in fitting real-world measurements for the expanded model.

To further improve the similarity of transient responses in Fig. 7, we may use weighting factors for quantities of interest (normalized in a standard way for power system applications).

Also, the agreement of transient responses in Fig. 7 may be improved with additional training of neural networks (especially for ***NN2***, ***NN3***, and ***NN4m***—see above reported MSEs of neural

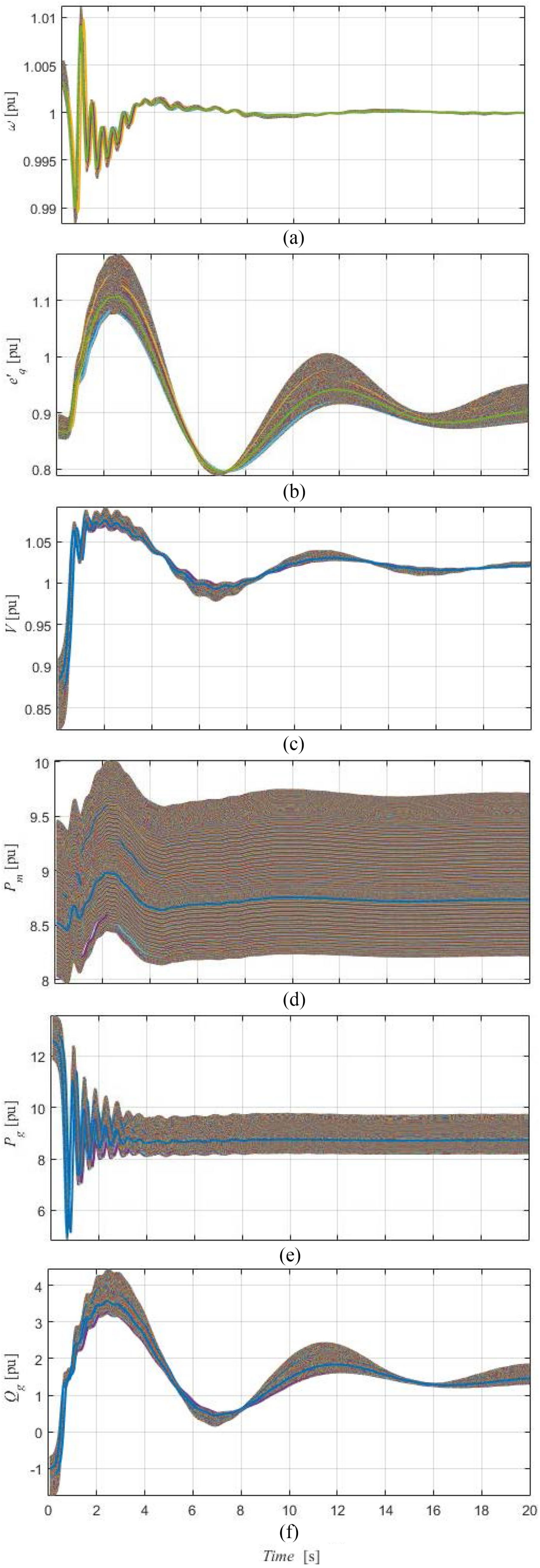


Fig. 5. Characteristic transient responses of state, algebraic, and measurement variables for expanded model (from training tensor $\boldsymbol{W}_e$). (a) Rotor speed. (b) Transient rotor voltage in q-axis. (c) Bus voltage magnitude. (d) Mechanical power. (e) Real (active) power. (f) Reactive power.

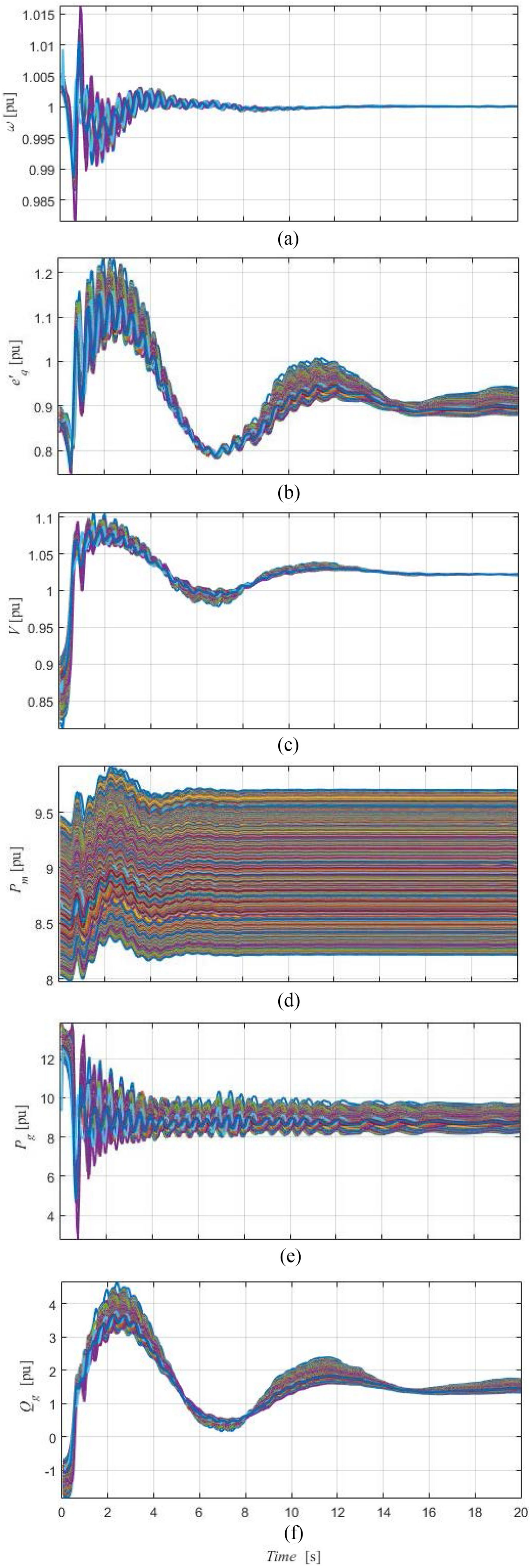


Fig. 6. Characteristic transient responses of state, algebraic, and measurements variables for practical model (from training tensor $\boldsymbol{W}_p$). (a) Rotor speed. (b) Transient rotor voltage in q-axis. (c) Bus voltage magnitude. (d) Mechanical power. (e) Real (active) power. (f) Reactive power.

TABLE II
BASIC INPUTS FOR TRAINING OF NEURAL NETWORKS IN FIG. 4

Neural network	Neurons in the input layer	Neurons in hidden layers	Neurons in the output layer	Size of the training pattern
NN1	$\mathrm{Size}\{\boldsymbol{y}\} = 7$	10, 10	$\mathrm{Size}\{\boldsymbol{x}_e\} = 6$	$\mathrm{N}_\mathrm{w}^\mathrm{com} \times \mathrm{N}_\mathrm{t}^\mathrm{com} = 554 \times 262$
NN2	$\mathrm{Size}\{\boldsymbol{y}\} = 7$	20, 20	$\mathrm{Size}\{\boldsymbol{x}_p\} + \mathrm{Size}\{\boldsymbol{z}_p\} = 7$	$\mathrm{N}_\mathrm{w}^\mathrm{com} \times \mathrm{N}_\mathrm{t}^\mathrm{com} = 554 \times 262$
NN3	$\mathrm{Size}\{\boldsymbol{x}_p\} + \mathrm{Size}\{\boldsymbol{z}_p\} + \mathrm{Size}\{\boldsymbol{y}_{xz,p}\} = 9$	20, 20	$\mathrm{Size}\{\boldsymbol{x}_p\} + \mathrm{Size}\{\boldsymbol{z}_p\} = 7$	$\mathrm{N}_\mathrm{w}^\mathrm{com} \times (\mathrm{N}_\mathrm{t}^\mathrm{com} - 1) = 554 \times 261$
NN4m; $\boldsymbol{m} = 1,2,\cdots,\mathrm{N}_\mathrm{m}$*	$\mathrm{N}_\mathrm{t}^\mathrm{com} = 262$	10, 10	312**	$\mathrm{N}_\mathrm{w}^\mathrm{com} = 554$

*Note that for ***NN4m*** there are $\boldsymbol{m} = 1, 2, \ldots, \mathrm{N}_\mathrm{m} = 7$ independent ***NN4m*** with a possibility for parallelized training

**For $r = 0.2$ is $\mathrm{Size}\{\boldsymbol{\lambda}\} + \mathrm{Size}\{\boldsymbol{\varphi}\} + \mathrm{Size}\ \{\boldsymbol{b}\} = 3 \times 2 \times 52 = 312$ (for complex variables $\boldsymbol{\lambda}$, $\boldsymbol{\varphi}$, and $\boldsymbol{b}$)

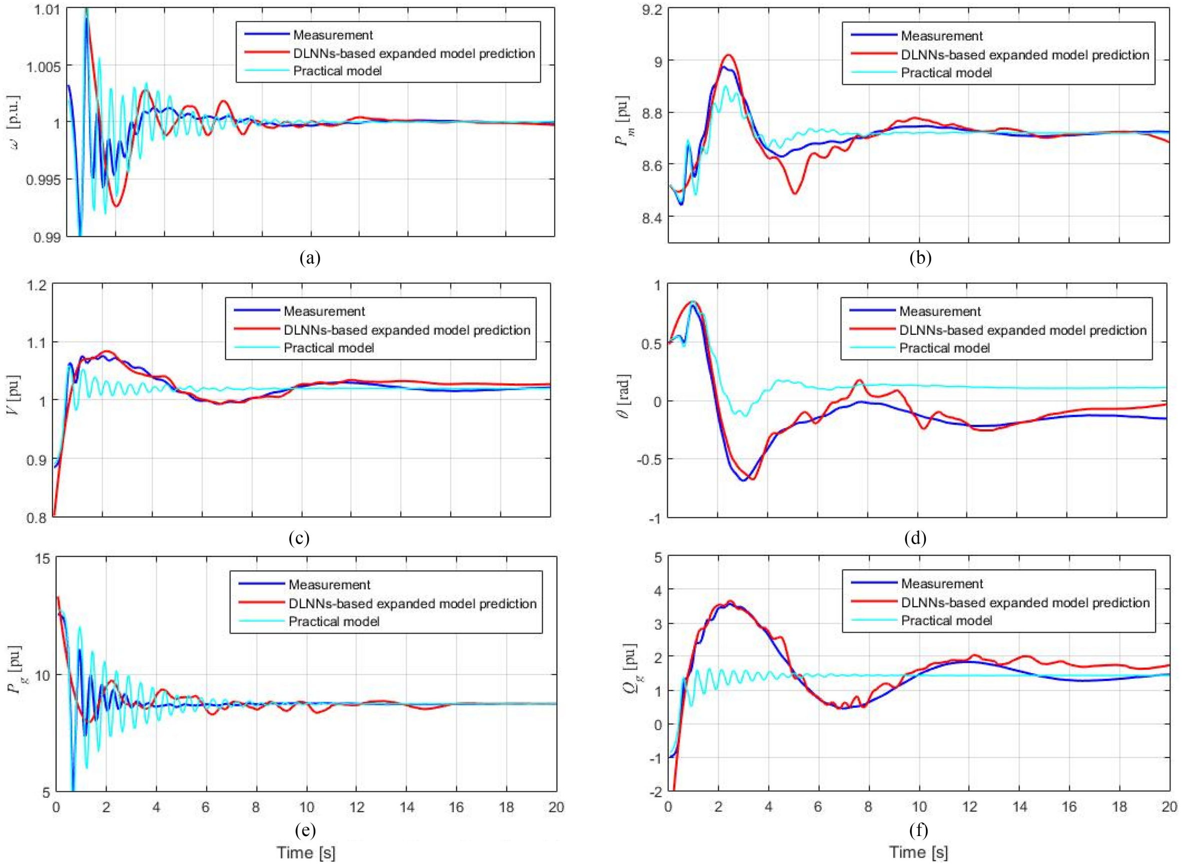


Fig. 7. Predicted transient responses of measurements (Verification phase), obtained by DNN-based gray-box (for verification initial condition). (a) Rotor speed. (b) Mechanical power. (c) Bus voltage magnitude. (d) Bus voltage angle. (e) Active power. (f) Reactive power.

networks training) or by increasing the number of inputs (for instance, by adding the time coordinate in data set for ***NN3***). Another possibility is modifying the number of Koopman modes retained (rank r).

Also, note that the differences in transient response are not the critical issue, because the conclusions about transient stability typically are: 1) critical fault clearing time, 2) maximum fault recovery voltage, and 3) fault ride through. In all cases given above, these metrics are correctly identified.

IX. CONCLUSION

Our focus in this article has been to explore the dynamic analyses (on the TSA example) in power systems from the standpoint of interleaving physics- and data-driven models. We combined compressed sensing, information geometry, Koopman operator, and gray-box hybrid integration, and show that our three-stage procedure holds promise in a realistic, multimachine benchmark example. Furthermore, with some straightforward modifications, our procedure may be applied to modeling and TSA of other dynamic components, such as, dynamic loads.

Presented results show that the reduced (equivalent) models sometimes have large errors in comparison with real-world measurements, meaning that the existing tools for power system dynamics are not satisfactory in all cases. To compensate for such errors, the integration with an additional data-driven model is necessary, both for real-time dynamics analysis or in long-term planning, such as TSA, dynamic voltage stability, angle stability, etc.

Input data to the designed DNN-based environment in the verification phase (real-time application) are only the initial conditions of local state and algebraic variables in PoC for the analyzed dynamic element, obtained from static state estimation. Local measurement-based TSA is beneficial in deregulated energy market environment, where data exchange is limited among different transmission owners and operators. The proposed model is, thus, applicable to both centralized and decentralized wide-area monitoring systems. We hope that this approach is relevant for other systems as well, and believe that broadening of the analyst's toolset is both timely and useful, given the rapid surge of data availability and size in modern engineered systems.

APPENDIX
SG INPUT DATA

$S_n = 2 \times 727.5\ \ \mathrm{MVA}; V_n = 21\ \ \mathrm{kV};\ \ f = 50\ \ \mathrm{Hz};\ \ \Omega_b = 2\ \pi f = 100\ \pi;\ \omega_0 = 1\ \mathrm{pu};\ 2H = 26.12\ \mathrm{MWs/MVA};\ D = 7.275\ \mathrm{pu};\ T'_{d0} = 6.5\ \mathrm{s};\ T'_{q0} = 0.31\ \mathrm{s};\ T''_{d0} = 0.02\ \mathrm{s};\ T''_{q0} = 0.042\ \mathrm{s};\ x_\ell = 0.02\,\mathrm{pu};\ r_a = 0\ \mathrm{pu};\ x_d = 0.352\ \mathrm{pu};\ x_q = 0.345\ \mathrm{pu};\ x'_d = 0.052\ \mathrm{pu};\ x'_q = 0.078\ \mathrm{pu};\ x''_d = 0.033\ \mathrm{pu};\ x''_q = 0.033\,\mathrm{pu}.$

REFERENCES

[1] A. T. Sarić, M. K. Transtrum, and A. M. Stanković, "Data-driven classification, reduction, parameter identification and state extension in hybrid power systems," *IEEE Trans. Power Syst.*, vol. 36, no. 3, pp. 2222–2233, May 2021.

[2] 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.

[3] 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.

[4] L. Wang, Q. Zhou, and S. Jin, "Physics-guided deep learning for power system state estimation," *J. Modern Power Syst. Clean Energy*, vol. 8, no. 4, pp. 607–615, Jul. 2020.

[5] X. Hu, H. Hu, S. Verma, and Z. L. Zhang, "Physics-guided deep neural networks for power flow analysis," *IEEE Trans. Power Syst.*, vol. 36, no. 3, pp. 2082–2092, May 2021.

[6] X. Lei, Z. Yang, J. Yu, J. Zhao, Q. Gao, and H. Yu, "Data-driven optimal power flow: A physics-informed machine learning approach," *IEEE Trans. Power Syst.*, vol. 36, no. 1, pp. 346–354, Jan. 2021.

[7] G. S. Misyris, A. Venzke, and S. Chatzivasileiadis, "Physics-informed neural networks for power systems," in *Proc. IEEE Power Energy Soc. Gen. Meeting*, Aug. 2020, pp. 2–6.

[8] Q. Wang, F. Li, Y. Tang, and Y. Xu, "Integrating model-driven and data-driven methods for power system frequency stability assessment and control," *IEEE Trans. Power Syst.*, vol. 34, no. 6, pp. 4557–4568, Nov. 2019.

[9] K. Chen, "Indirect PCA dimensionality reduction based machine learning algorithms for power system transient stability assessment," in *Proc. Innov. Smart Grid Technol. Asia*, May 2019, pp. 4175–4179.

[10] F. Li, Q. Wang, Y. Tang, Y. Xu, and J. Dang, "Hybrid analytical and data-driven modeling based instance-transfer method for power system online transient stability assessment," *CSEE J. Power Energy Syst*., Apr. 2021.

[11] T. Huang, S. Gao, and L. Xie, "A neural lyapunov approach to transient stability assessment of power electronics-interfaced networked microgrids," *IEEE Trans. Smart Grid*, vol. 13, no. 1, pp. 106–118, Jan. 2022.

[12] $* * *$, *TOP-010-1(i) – Real-time Reliability Monitoring and Analysis Capabilities to Support Reliable System Operations and Compliance Implementation Guidance Real-Time Assessment Quality of Analysis*. Cambridge, U.K.: North American Reliability Corporation (NERC), May 2019.

[13] H. Khoshkhoo, S. Yari, A. Pouryekta, V. K. Ramachandaramurthy, and J. M. Guerrero, "A remedial action scheme to prevent mid/long-term voltage instabilities," *IEEE Syst. J.*, vol. 15, no. 1, pp. 923–934, Mar. 2021.

[14] G. Wu, M. Li, and Z. S. Li, "A stochastic modeling approach for cascading failures in cyber-physical power systems," *IEEE Syst. J.*, [Online]. Available: https://ieeexplore.ieee.org/document/9411790/

[15] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, May 2015.

[16] S. L. Brunton, J. L. Proctor, and J. N. Kutz, "Discovering governing equations from data by sparse identification of nonlinear dynamical systems," *Proc. Nat. Acad. Sci. USA*, vol. 113, no. 15, pp. 3932–3937, Mar. 2016.

[17] A. M. Stanković, A. A. Sarić, A. T. Sarić, and M. K. Transtrum, "Data-driven symbolic regression for identification of nonlinear dynamics in power systems," in *Proc. IEEE Power Energy Soc. Gen. Meeting*, 2020, pp. 2–6.

[18] O. Ajala, A. Dominguez-Garcia, P. Sauer, and D. Liberzon, "A library of second-order models for synchronous machines," *IEEE Trans. Power Syst.*, vol. 35, no. 6, pp. 4803–4814, Nov. 2020.

[19] E. V. Filho and P. L. dos Santos, "A dynamic mode decomposition approach with hankel blocks to forecast multi-channel temporal series," *IEEE Control Syst. Lett.*, vol. 3, no. 3, pp. 739–744, Jul. 2019.

[20] P. Sharma, B. Huang, U. Vaidya, and V. Ajjarapu, "Data-driven identification and prediction of power system dynamics using linear operators," in *Proc. IEEE Power Energy Soc. Gen. Meeting*, Apr.–Aug. 2019, pp. 4–8.

[21] J. J. Ramos and J. N. Kutz, "Dynamic mode decomposition and sparse measurements for characterization and monitoring of power system disturbances," 2019, *arXiv:1906.03544*.

[22] Y. Susuki and I. Mezić, "Nonlinear koopman modes and power system stability assessment without models," *IEEE Trans. Power Syst.*, vol. 29, no. 2, pp. 899–907, Mar. 2014.

[23] A. M. Stanković, A. A. Sarić, A. T. Sarić, and M. K. Transtrum, "Interleaving physics- and data-driven models for power system transient dynamics," *Elect. Power Syst. Res.*, vol. 189, Dec. 2020.

[24] E. Yeung, S. Kundu, and N. O. Hodas, "Learning deep neural network representation for koopman operators of nonlinear dynamical systems," in *Proc. Amer. Control Conf.*, Jul. 2019, pp. 10–12.

[25] B. Lush, J. N. Kutz, and S. L. Brunton, "Deep learning for universal linear embeddings of nonlinear dynamics," *Nature Commun.*, vol. 9, Nov. 2018, Art. no. 4950.

[26] S. L. Brunton, J. L. Proctor, and J. N. Kutz, "Compressive sampling and dynamic mode decomposition," *J. Comput. Dyn.*, vol. 2, no. 2, pp. 165–191, 2015. [Online]. Available: https://arxiv.org/abs/1312.5186

[27] A. Majamder, *Compressed Sensing for Engineers*. Boca Raton, FL: CRC Press, 2018.

[28] E. Candés and J. Romberg, "ℓ_1-magic: Recovery of sparse signals via convex programming," 2005. [Online]. Available: http://brainimaging.waisman.wisc.edu/~chung/BIA/download/matlab.v1/l1magic-1.1/l1magic_notes.pdf

[29] H. Arbabi and I. Mezić, "Ergodic theory, dynamic mode decomposition and computation of spectral properties of the koopman operator," *SIAM J. Appl. Dynamical Syst.*, vol. 16, no. 4, pp. 2096–2126, 2017.

[30] J. H. Tu *et al.*, "On dynamic mode decomposition: Theory and applications," *Comput. Dyn.*, vol. 1, no. 2, pp. 391–421, 2014.

[31] K. K. Chen, J. H. Tu, and C. W. Rowley, "Variants of dynamic mode decomposition: Boundary condition, koopman, and fourier analyses," *Nonlinear Sci.*, vol. 22, no. 6, pp. 887–915, Dec. 2012.

[32] P. J. Schmid, "Dynamic mode decomposition for numerical and experimental data," *Fluid Mech.*, vol. 656, pp. 5–28, Aug. 2010.

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

[34] J. N. Kutz, X. Fu, S. L. Brunton, and N. B. Erichson, "Multi-resolution dynamic mode decomposition for foreground/background separation and object tracking," in *Proc. IEEE Int. Conf. Comput. Vis. Workshop*, 2015, pp. 921–929.

[35] M. Gavish and D. L. Donoho, "The optimal hard threshold for singular value is $4/\sqrt{3}$," *IEEE Trans. Inf. Theory,* vol. 60, no. 8, pp. 5040–5053, Aug. 2014.

[36] Tufts Box. [Online]. Available: https://tufts.app.box.com/folder/145176754974

[37] F. Milano, *Power System Modelling and Scripting*. New York, NY, USA: Springer-Verlag, 2010.

[38] E. Ghahremani and I. Kamwa, "Local and wide-area PMU-based decentralized dynamic state estimation in multi-machine power systems," *IEEE Trans. Power Syst.*, vol. 31, no. 1, pp. 547–562, Feb. 2015.

[39] E. Ghahremani and I. Kamwa, "Dynamic state estimation in a power system by applying the extended Kalman filter with unknown inputs to phasor measurements," *IEEE Trans. Power Syst.*, vol. 26, no. 4, pp. 2556–2566, Jul. 2011.

[40] C. Wang, X. Fu, P. Li, J. Wu, and L. Wang, "Multiscale simulation of power system transients based on the matrix exponential function," *IEEE Trans. Power Syst.*, vol. 32, no. 3, pp. 1913–1926, May 2017.