Bayesian Physics-informed Neural Networks for system identification of inverter-dominated power systems

Stock, Simon; Babazadeh, Davood; Becker, Christian; Chatzivasileiadis, Spyros

Published in:
Electric Power Systems Research

Link to article, DOI:
10.1016/j.epsr.2024.110860

Publication date:
2024

Document Version
Publisher's PDF, also known as Version of record

Electric Power Systems Research 235 (2024) 110860

Contents lists available at ScienceDirect

Electric Power Systems Research

journal homepage: www.elsevier.com/locate/epsr

Bayesian Physics-informed Neural Networks for system identification of inverter-dominated power systems

Simon Stock [a,*], Davood Babazadeh [a], Christian Becker [a], Spyros Chatzivasileiadis [b]

[a] Institute of Power and Energy Technology, Hamburg University of Technology, Hamburg, Germany
[b] Department of Wind and Energy Systems, Technical University of Denmark, Kgs. Lyngby, Denmark

ARTICLE INFO

Keywords:
Bayesian Physics-informed Neural Networks
System identification
Inverter-dominated power systems
Machine learning

ABSTRACT

While the uncertainty in generation and demand increases, accurately estimating the dynamic characteristics of power systems becomes crucial for employing the appropriate control actions to maintain their stability. In our previous work, we have shown that Bayesian Physics-informed Neural Networks (BPINNs) outperform conventional system identification methods in identifying the power system dynamic behavior based on noisy data. This paper takes the next natural step and addresses the more significant challenge, exploring how BPINN performs in estimating power system dynamics under increasing uncertainty from many Inverter-based Resources (IBRs) connected to the grid. These introduce a different type of uncertainty, compared to noise. The BPINN combines the advantages of Physics-informed Neural Networks (PINNs), such as inverse problem applicability, with Bayesian approaches for uncertainty quantification. We explore the BPINN performance on a wide range of systems, starting from a single machine infinite bus (SMIB) system and 3-bus system to extract important insights, to the 14-bus CIGRE distribution grid, and the large IEEE 118-bus system. We also investigate approaches that can accelerate the BPINN training, such as pretraining and transfer learning. Throughout this paper, we show that in presence of uncertainty, the BPINN achieves orders of magnitude lower errors than the widely popular method for system identification SINDy and significantly lower errors than PINN, while transfer learning helps reduce training time by up to 75%.

1. Introduction

The ongoing integration of Inverter-based Resources (IBRs) into the energy system leads to significant changes in frequency dynamics, since they do not show the same characteristics as conventional synchronous generators. Additionally, the volatile behavior of wind power plants and PV causes a fluctuating infeed of power that increases uncertainty in grid operation. For that reason, system operators appear to lack system awareness in case of high penetration of renewables. Nevertheless, the introduction of fast and distributed measurement devices, namely phasor-measurement-units (PMUs), enables the utilization of data-driven approaches for identification of system dynamics.

Various approaches have been introduced, such as filter-based techniques, e.g. Kalman filtering [1], Koopman theory [2] or parsimonious approaches, for example Sparse Identification of Nonlinear Dynamics (SINDy) [3]. Many of them have shown vulnerability to uncertainty in the data, resulting in inaccurate estimates. Machine learning and, more recently, hybrid approaches that combine the strengths of machine learning with physics-based models have been introduced to solve this problem, for example the Physics-informed Neural Networks (PINNs) [4,5]. Besides the PINN [6], which utilizes a physics-based DAE system to augment the loss function, there have been various other approaches. For example, multiple techniques have been proposed that use the graph structure of power system matrices, such as [7–9]. However, these graph-based approaches cannot be applied to the system identification problem at hand, since most of them do not focus on the dynamic characteristics of the power systems. However, Neural Networks (NNs) and PINNs do not inherently quantify the experienced uncertainty and therefore lack a confidence indication for their estimate. In Bayesian techniques, the estimate is augmented with such a confidence measure by design [10].

In that sense, the Bayesian Physics-informed Neural Network (BPINN) has been introduced which combines the PINN and Bayesian techniques [11]. In power systems, it has shown robustness against uncertainty from noisy data for system identification [12], outperforming established approaches, such as SINDy. This type of uncertainty is commonly known as aleatoric uncertainty. In this paper, we further evaluate the Bayesian Physics-informed Neural Network (BPINN) as follows: First, we investigate the BPINN performance for estimating the

* Corresponding author.
E-mail address: simon.stock@tuhh.de (S. Stock).

https://doi.org/10.1016/j.epsr.2024.110860
Received 20 March 2024; Received in revised form 4 June 2024; Accepted 1 July 2024
Available online 17 July 2024

frequency dynamics of an inverter-dominated grid. IBRs lead to model uncertainties in system identification, which are commonly known as epistemic uncertainty. Second, we seek to transfer previously learned knowledge in BPINN training. After pretraining on a single machine infinite bus (SMIB) system, we transfer the knowledge, i.e. use the BPINN with pretrained weights and biases, and train on a larger system in order to reduce the required data and training iterations.

Most Bayesian approaches require informative prior knowledge about the inferred system parameters [10]. These are expected to change frequently for inverter-dominated power systems, thus, we focus on weakly-informative priors. In contrast to informative priors, weakly-informative priors can be generally applied to the whole range of system parameters.

The rest of this paper is structured as follows: We start by introducing the methodology of the BPINN in Section 2, highlight the similarities to the PINN, and detail its uncertainty quantification capabilities. We then specify the weakly-informative priors and simulation parameters. In Section 3, the IBR and generator models are presented and the four grid models are specified. Section 4 discusses the results of the system identification and transfer learning and compares it to the PINN and SINDy. Section 5 concludes this paper.

2. Methodology

We start this section by revisiting the NN and extend it to the PINN formulation. Based on that, we present the BPINN and its uncertainty quantification capabilities.

Let us assume a dynamic model described by the following set of differential equations:

$$\dot{\boldsymbol{x}} = f(\boldsymbol{x}, \boldsymbol{u}; \lambda) \tag{1}$$

with solution $\boldsymbol{x}(t, \boldsymbol{u})$, $\boldsymbol{x}$ representing the states and $\boldsymbol{u}$ the inputs of the system. Variable λ describes the system parameters, e.g. the damping constant of a mass–spring oscillator, and operator f maps the system parameters to the states.

2.1. Neural networks

NNs can generally be used as function approximators for a variety of problems. For dynamic systems, they can serve as a surrogate model $g(t)$ mapping a time-dependent input vector to the target trajectory of the states $\boldsymbol{x}$:

$$g(t; \boldsymbol{\Theta}) = \hat{\boldsymbol{x}}(t; \boldsymbol{\Theta}) \approx \boldsymbol{x}(t, \boldsymbol{u}; \boldsymbol{x_0}, \lambda). \tag{2}$$

where $\boldsymbol{x_0}$ describes the initial state of the system and $\boldsymbol{\Theta}$ the Neural Network parameters, i.e. the NN weights and biases. A set of data $\mathcal{D}$ with $\mathcal{D} = \{\boldsymbol{x}^{(i)}, \boldsymbol{u}^{(i)}\}_{i=1}^{N_z}$ can be used to determine the surrogate model parameters $\boldsymbol{\Theta}$ through training. The distance between the estimate $\hat{\boldsymbol{x}}(t)$ and the target trajectory is minimized by updating the NN parameters $\boldsymbol{\Theta}$. In (3), it is formulated as an optimization problem using the root-mean-squared error as the distance measure. This NN training procedure is a supervised learning problem, i.e. it requires a fully labeled dataset $\mathcal{D}$ including all true values $\boldsymbol{x}$.

$$\min_{\boldsymbol{\Theta}} \mathcal{L}_D = \min_{\boldsymbol{\Theta}} \frac{1}{N_z} \sum_{i=1}^{N_z} \sqrt{(\hat{\boldsymbol{x}}^{(i)}(\boldsymbol{\Theta}) - \boldsymbol{x}^{(i)})^2} \tag{3}$$

The formulation (3) strives to find the optimal model parameters $\boldsymbol{\Theta}$. However, it is not able to obtain the system parameters λ. An inverse problem has to be formulated to determine the system parameters λ as well. To this end, the PINN has been proposed [6]. PINNs utilize a physics regularization term, as follows:

$$h(\boldsymbol{\Theta}, \hat{\lambda}) = \frac{d}{dt}\hat{\boldsymbol{x}} - f(\hat{\boldsymbol{x}}, \boldsymbol{u}; \hat{\lambda}) \overset{!}{=} 0. \tag{4}$$

The function h represents a residual that is zero if the state estimate $\hat{\boldsymbol{x}}$ and parameter estimates $\hat{\lambda}$ are accurate. It is based on the DAE that describes the dynamic model (1) or an approximation of that and augments the loss function (3) in the form

$$\min_{\boldsymbol{\Theta}} \mathcal{L}_D + \frac{1}{N} \sum_{i=1}^{N} \sqrt{h^{(i)2}} \tag{5}$$

The parameters $\boldsymbol{\Theta}$ can now be tuned to find the state estimates $\hat{\boldsymbol{x}}$ and system parameter estimate $\hat{\lambda}$ collaboratively. This formulation does not require labeled data for λ, while it still necessitates labels for $\boldsymbol{x}$.

The evaluation of (4) can be extended with additional data that are not in the initial data $\mathcal{D}$. The NN can be evaluated at every time point t to generate N_c so-called collocation points, so that the total number of training points is $N = N_z + N_c$. The N_c collocation points can support the physics regularization in (4). In conclusion, the PINN enables estimating the state trajectory $\boldsymbol{x}$ and system parameters λ at the same time based on given data $\mathcal{D}$, however, it does not indicate the uncertainty of the estimates.

2.2. Bayesian PINN

Bayesian frameworks have been introduced to deep learning techniques to quantify the uncertainty in data and models [13]. The literature mostly distinguishes between two types of uncertainty: first, aleatoric uncertainty, which represents noise inherent in the observed data and second, epistemic uncertainty, which describes uncertainty in the model.

Uncertainty quantification. Aleatoric uncertainty is commonly assumed to be Gaussian distributed. An artificial set of observed noisy data $\mathcal{D}$ can be created with a deterministic process, such as (1), giving the mean and additive noise provided by a covariance matrix $\Sigma_x = \sigma_x^2 \boldsymbol{I}$. This type of uncertainty can potentially be quantified by extending (3) as follows [13]:

$$\min_{\boldsymbol{\Theta}} \frac{1}{N} \sum_{i=1}^{N} \frac{1}{2\sigma_x^{(i)2}} \sqrt{(\hat{\boldsymbol{x}}^{(i)}(\boldsymbol{\Theta}) - \boldsymbol{x}_{\text{true}}^{(i)})^2} + \frac{1}{2} \log \sigma_x^{(i)2}. \tag{6}$$

Note that this formulation does not lead to a Bayesian NN, which means that we have to find a single value for each NN parameter Θ.

Bayesian Neural Networks (BNNs) were introduced to address the problem of epistemic uncertainty, resulting from the model, about three decades ago [14]. A probability distribution $p(\boldsymbol{\Theta})$ is placed on the model parameters $\boldsymbol{\Theta}$ based on prior beliefs. In training, their posterior distributions $p(\boldsymbol{\Theta}|D)$ are inferred using Bayes' rule and the data $\mathcal{D}$. Subsequently, the BNN can be envisioned as a family of models that incorporates the plausible set of parameters. The posterior distribution of model parameters $\boldsymbol{\Theta}$ can be obtained based on

$$p(\boldsymbol{\Theta}|D) = \frac{p_D(D|\boldsymbol{\Theta}) p(\boldsymbol{\Theta})}{p(D)} \tag{7}$$

using a Bayesian inference algorithm. Samples can be pulled from the inferred parameter distribution $\boldsymbol{\Theta}^* \sim p(\boldsymbol{\Theta}|D)$. These can be used to calculate the mean of the distribution of estimated states by

$$\mathbb{E}[\hat{\boldsymbol{x}}|\boldsymbol{u}, D] \approx \frac{1}{T} \sum_{i}^{T} \hat{\boldsymbol{x}}_{\Theta_i^*}(u) := \overline{\boldsymbol{x}}_D(\boldsymbol{u}) \tag{8}$$

and the aleatoric uncertainty as $\mathbb{E}_{\Theta|D}[\text{Var}(\hat{\boldsymbol{x}}|\boldsymbol{u}, \boldsymbol{\Theta})]$. The aleatoric uncertainty consists of the expected variance of $\hat{\boldsymbol{x}}$, in contrast, the epistemic uncertainty can be formulated as the variance of the expected value of $\hat{\boldsymbol{x}}$. This gives us the total uncertainty [15]:

$$\text{Var}(\hat{\boldsymbol{x}}|\boldsymbol{u}, D) = \mathbb{E}_{\boldsymbol{\Theta}|D}[\text{Var}(\hat{\boldsymbol{x}}|\boldsymbol{u}, \boldsymbol{\Theta})] + \text{Var}_{\boldsymbol{\Theta}|D}(\mathbb{E}[\hat{\boldsymbol{x}}|\boldsymbol{u}, \boldsymbol{\Theta}]). \tag{9}$$

The formulation assumes that the distribution $p(\hat{\boldsymbol{x}}|D)$ follows the same distribution as the observed data $\mathcal{D}$. Although we can disassemble (9) into its aleatoric and epistemic elements in the form of equations, the BNN cannot distinguish between the sources of uncertainty by design, so it only provides one uncertainty measure.

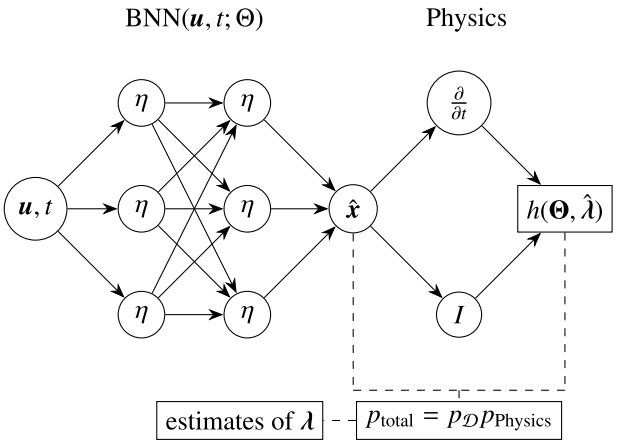

Fig. 1. Bayesian Physics-informed Neural Network schematic with nonlinear activation η.

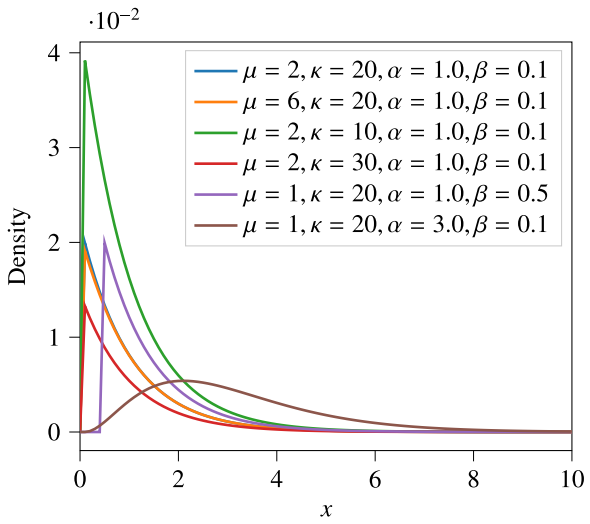

Fig. 2. Exemplary normal-gamma PDFs for different parameters μ, κ, α, β.

The above formulations enable the BNN to discover the state estimates $\hat{\boldsymbol{x}}$ considering the aleatoric and epistemic uncertainty based on:

$$p_D(\mathcal{D}|\boldsymbol{\Theta}) = \prod_i^N \frac{1}{\sqrt{2\pi\sigma_x^{(i)2}}} \exp\left(-\frac{(\hat{\boldsymbol{x}}^{(i)}(\boldsymbol{\Theta}) - \boldsymbol{x}_{\text{true}}^{(i)})^2}{2\sigma_x^{(i)2}}\right). \tag{10}$$

Please note, that we do not assume the fidelity of the data to be known a priori, that would require additional information to the data itself. The σ_x has to be determined during training.

The formulation (10) is solely applicable to forward problems, so we are only able to estimate the system states $\boldsymbol{x}$. Simultaneous estimation of the system parameters λ requires the extension of the formulation. A physical regularization similar to (4) is introduced, making the BNN a BPINN [11]. This leads to the following equation:

$$p_{\text{total}}(\mathcal{D}|\boldsymbol{\Theta}, \lambda) = p_D(\mathcal{D}|\boldsymbol{\Theta}) \prod_i^N \frac{1}{\sqrt{2\pi\sigma_h^{(i)2}}} \exp\left(-\frac{(h^{(i)}(\boldsymbol{\Theta}, \lambda))^2}{2\sigma_h^{(i)2}}\right). \tag{11}$$

The BPINN structure is illustrated in Fig. 1. Based on (11), the joint posterior of the BNN parameters $\boldsymbol{\Theta}$ and system parameters λ can be determined following Bayes' theorem with prior distributions $p(\boldsymbol{\Theta})$ and $p(\lambda)$.

$$p(\boldsymbol{\Theta}, \lambda|D) = \frac{p_{\text{total}}(\mathcal{D}|\boldsymbol{\Theta}, \lambda) p(\boldsymbol{\Theta}, \lambda)}{p(D)}. \tag{12}$$

BPINN training. We use Variational Inference (VI) to find the joint posterior of weights and biases $\boldsymbol{\Theta}$ and system parameters λ, which provides a computationally efficient formulation, that can be solved with common libraries for NN training. In this paper, we specifically rely on the Stein Variational Gradient Descend (SVGD) algorithm to solve this task, for more information, the interested reader can refer to [16].

In conclusion, the BPINN estimates the system states $\hat{\boldsymbol{x}}$ and system parameters λ, while inherently indicating the uncertainty of the estimated value considering the aleatoric and epistemic uncertainty.

Priors for BPINNs. The calculation of (11) requires to set a distribution $p(\boldsymbol{\Theta}, \lambda)$ based on prior beliefs. A commonly used prior $\boldsymbol{\Theta}$ for BNN parameters is a Gaussian distribution with zero mean, standard deviation $\sigma_{w,l} = 1$ and $\sigma_{b,l} = 1$ for weights and biases w_l and b_l. When the same prior is used for system parameters λ, prior knowledge of the range of λ can be required. For inverter-dominated power systems, the system parameters are expected to change frequently. Hence, it is difficult to constantly update the prior beliefs. In this paper, we use a generic prior for the system parameters λ that does not require informative knowledge. The so-called weakly-informative priors seek to include as little information as possible. They are based on scale mixtures of normals [17]. Specifically, we apply the normal-gamma distribution, whose probability density function can be expressed as:

$$\lambda_{prior} \sim \mathcal{N}(\mu, \kappa/\iota), \iota \sim \Gamma(\beta, \alpha). \tag{13}$$

The normal-gamma has a spike close to zero, similar to the Laplace distribution, and thus allows stronger regularization than the normal distribution. More importantly, the parameters α and β can be used to control the information content of the distribution [17]. Exemplary probability density functions for a normal-gamma distribution are shown in Fig. 2. We briefly explore these parameters in Section 4 to choose the distribution parameters for this paper.

2.3. System identification of power systems

In this paper, we estimate the dynamic frequency behavior of an inverter-dominated power system. This can generally be approximated by a SMIB representation [18]

$$\dot{\delta} = \Delta\omega \tag{14}$$

$$\Delta\dot{\omega} = \frac{1}{m}(P_m - d\Delta\omega - B\sin(\delta)). \tag{15}$$

Variable m represents the system inertia, d the damping, B the susceptance and P_m the mechanical power. The voltage is assumed to be one. The states of the system are the angle and the frequency deviation $\boldsymbol{x} = \{\delta, \Delta\omega\}$. The SMIB representation neglects various effects when representing a more complex system with numerous IBRs. However, we will utilize it for the physics regularization in (11) throughout the entire paper, since it still properly describes the general frequency behavior. Consequently, BPINN provides two distribution estimates for the system states $\boldsymbol{x}$ and three distributions for the system parameters $\lambda = \{m, d, B\}$. These can represent a larger system, such as 118-bus, in an aggregated form.

Finally, the BPINN parameter estimates $\hat{\lambda} = \{\hat{m}, \hat{d}, \hat{B}\}$ can be obtained by taking the mean of the posterior distribution and a measure of confidence is given based on the posterior variance, as described in Section 2.2. The latter indicates if BPINN is confident about the given estimate.

All simulations start from an unperturbed state, so $\dot{\boldsymbol{x}} = 0$. We perturb the system at $t = 0$ by applying a constant step in P_m, so $P_{m,step} = -0.1$ p.u. and collect the resulting state trajectories $\boldsymbol{x}$.

For all tests, a BPINN with 20 neurons, 1 hidden layer, and a standard trajectory length of $T = 5$ s was used at a sampling frequency

of 20 Hz, giving $N_z = 100$ samples. The inputs of the BPINN are t and P_m. The surrogate model of BPINN, the BNN, produces state trajectory estimates $\hat{\boldsymbol{x}}$. In addition, the physics part, based on (4), provides the system parameter estimates $\hat{\boldsymbol{\lambda}}$.

The BPINN was implemented in Python using packages *pytorch* and *numpyro*.

2.4. SINDy algorithm

The SINDy algorithm, proposed in [3], is part of the recently popular parsimonious approaches. These focus on the active terms in a given set of differential equations to reduce the computational effort. A library of candidate functions is defined $\boldsymbol{\zeta}(\boldsymbol{x})$, which can be polynomial combinations of the system states $\boldsymbol{x}$. These are used to formulate a set of differential equations that represent the behavior of the system. SINDy strives to reduce the number of equations and identify a sparse system representation $\boldsymbol{\zeta}(\boldsymbol{x})\boldsymbol{\Xi}$, with $\boldsymbol{\Xi}$ being the coefficients. Linear regression is utilized and the number of active equations is penalized through an additional term:

$$\underset{\boldsymbol{\Xi}}{\operatorname{argmin}} \|\dot{\boldsymbol{x}} - \boldsymbol{\zeta}(\boldsymbol{x})\boldsymbol{\Xi}\|_2 + \nu\|\boldsymbol{\Xi}\|_1. \tag{16}$$

The $\dot{\boldsymbol{x}}$ represents the derivatives of the states similar to previous formulations, while $\boldsymbol{\zeta}(\boldsymbol{x})\boldsymbol{\Xi}$ is the dynamic system with candidate functions $\boldsymbol{\zeta}(\boldsymbol{x})$. In the estimation phase, SINDy aims to find the coefficient vector $\boldsymbol{\Xi}$ that minimizes the first term of the equation. These coefficients are additionally used in a regularizing term $\nu\|\boldsymbol{\Xi}\|_1$ to allow for a sparse model. The candidate functions are known in (15), since we follow the SMIB representation. For that reason, the number of candidate functions is fixed in SINDy, which allows us to neglect the penalizing term $\boldsymbol{\zeta}$. For all tests the *PySINDy* library was used [19].

3. Case study

In previous work [12], we compared the system identification capabilities of BPINN, PINN and SINDy under aleatoric uncertainty resulting from noise in the data. This paper focuses on epistemic uncertainty arising from IBRs. To this end, we do not add aleatoric uncertainty to our training and estimation data, although there might be combined effects of aleatoric and epistemic uncertainty on the BPINN. However, these have been found to be small, thus, we focus solely on epistemic uncertainty in the present paper. We first explore the performance using data from the SMIB system. Three different dynamic situations are studied that serve as a baseline for the paper. Second, we implement a 3-bus system with one synchronous generator and two IBRs. This system aims to support a general understanding of the behavior of all algorithms at different levels of IBR penetration. We do not vary the inverter parameters here, thus, the same system with varying shares of synchronous behavior is observed. Third, data from the CIGRE 14-bus system are collected. This system represents a distribution system, with vast penetration of IBRs and a superordinate transmission system. We test different parameters for the transmission system and also randomly vary the inverter parameters J_c and d_c by ±20% following a uniform distribution. Fourth, we use data from the IEEE 118-bus system. We keep the synchronous generators and inverter parameters fixed and vary the number of inverter-coupled and synchronous generators to achieve different levels of IBR penetration.

Note that we estimate a single set of parameters m, d, B based on (15) for each grid. These parameters approximate the overall system dynamics.

3.1. Network models

The networks, 3-bus, 14-bus, 118-bus, consist of the corresponding number of nodes n_n nodes and n_b branches. The SMIB system is chosen as a baseline as described in the previous section. We use the SMIB in (15), so the SMIB regression formulation accurately represents the behavior of the system. We vary the generator parameters as shown in Table 1. The loading and operation conditions remain the same for all scenarios, since we exclusively use static loads. The self-regulating effect in the ENTSO-E system is only around 1%/Hz and is therefore neglected in this paper [20].

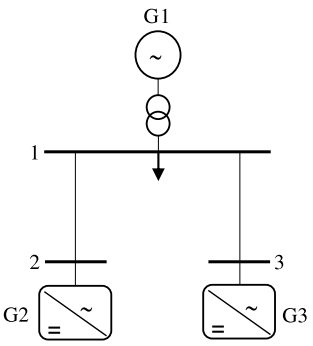

Fig. 3. Diagram of 3-bus system with IBRs.

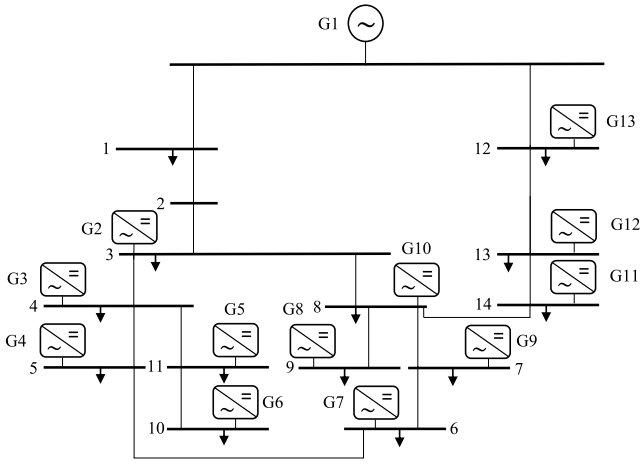

Fig. 4. CIGRE 14-bus MV system schematic with IBRs.

The influence of IBRs is investigated in more detail using a simple 3-bus system. This grid is a consecutive step from the SMIB formulation to a simple system representation that respects the influences of IBRs. We create a system with one synchronous generator and two IBRs, as shown in Fig. 3. Different dynamic situations are simulated by changing the parameters of the synchronous generator as shown in Table 1. We still use the same regression formulation (15) as before, which is no longer accurate due to the IBRs influence.

After exploring the BPINN performance on a small-scale system, we perform parameter estimations on the CIGRE 14-bus MV system. The system includes 12 IBRs. The structure of the system is shown in Fig. 4. We vary the dynamics by changing the parameters of the generator, following Table 1, and additionally randomly change the J_c and d_c parameters for each of the inverters by 20% around the values given in Table 2.

Finally, we test all algorithms on the IEEE 118-bus system. This system does not have any buses intended for IBRs, however, we substituted synchronous generators with IBRs, as illustrated in Fig. 5. Different dynamic situations, slow and fast, are created by setting all synchronous generators and IBRs to fixed parameters and altering the total number of IBRs and generators. In the fast dynamics scenario, only G1 is a synchronous generator, all other generator buses are equipped with IBRs. From there on, generator buses G1, G7, G5 are provided

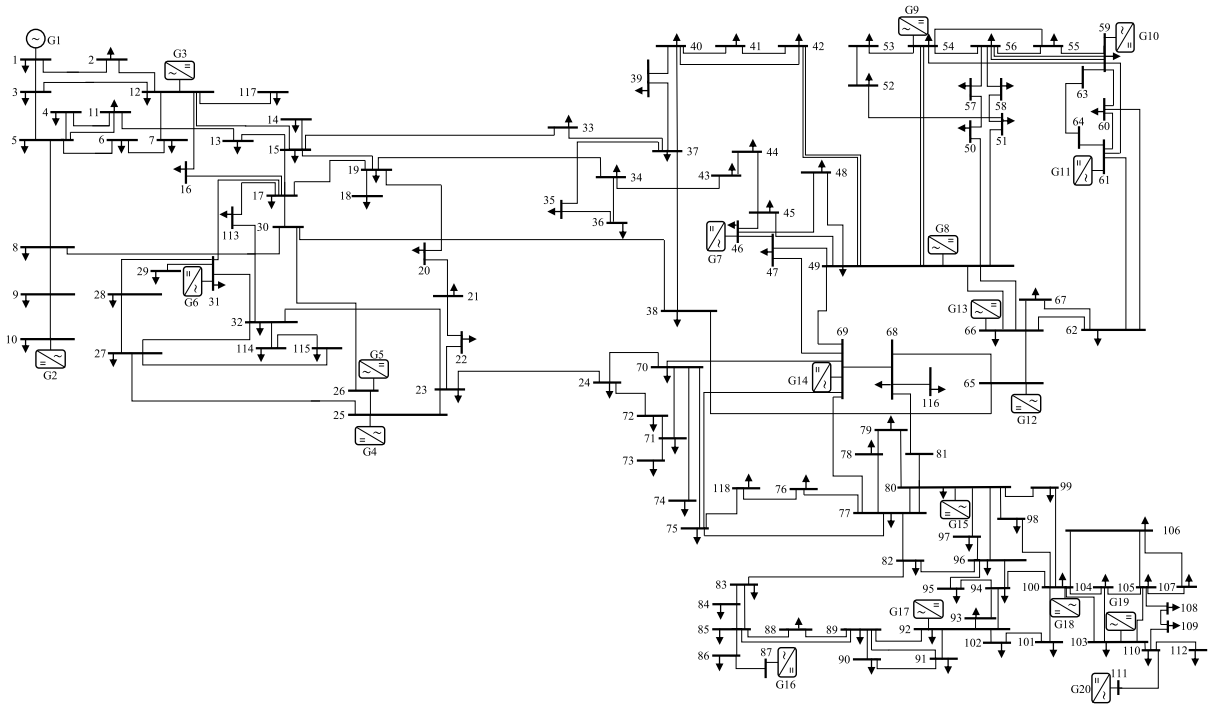

Fig. 5. Diagram of IEEE 118-bus with IBRs.

Table 1
Base evaluation scenarios generator parameters.

Scenario	m_{gen} in p.u.	d_{gen} in p.u.
Fast dynamics	1.1	0.8
Medium dynamics	1.5	1.2
Slow dynamics	2.1	1.8

with synchronous generators in the medium dynamics scenario. The slow dynamics scenario consists of synchronous generators connected to buses G1, G7, G5, G17, G14, G18, while all other generator buses are equipped with IBRs.

We start all simulation at a stable operating point, so $\dot{\boldsymbol{x}} = 0$ at $t = 0.0$ s as described in Section 2.3 and perturb the mechanical power of G1. All systems, the SMIB, 3-bus, 14-bus, 118-bus are perturbed similarly . The trajectories of $\Delta\omega, \delta$ are collected at all synchronous generators if multiple synchronous generators are active. In that case, a common frequency is calculated for the overall system at the center of inertia, thus, the trajectories from the individual generators are weighted by the individual and the total inertia constants as proposed in [21].

3.2. Inverter model

We model IBRs as a battery connected via a synchronverter [22,23]. A simplified diagram of the electrical and control parts is shown in Fig. 6.

For simplification, we neglect the DC side and switching, and focus solely on calculation of the synchronverter control, coupled with an RLC filter, R_f, L_f, C_f, and transformer, R_T, L_T. The power side of the inverter is modeled as follows

$$\dot{\underline{I}}_{RL} = \frac{1}{L_f}(\underline{E} - \underline{V}_f - R_f \underline{I}_{RL}) \tag{17}$$

$$\dot{\underline{V}}_f = \frac{1}{C_f}(\underline{I}_{RL} - \underline{I}_g) \tag{18}$$

$$\dot{\underline{I}}_g = \frac{1}{L_T}(\underline{V}_f - \underline{V}_T - R_T \underline{I}_g) \tag{19}$$

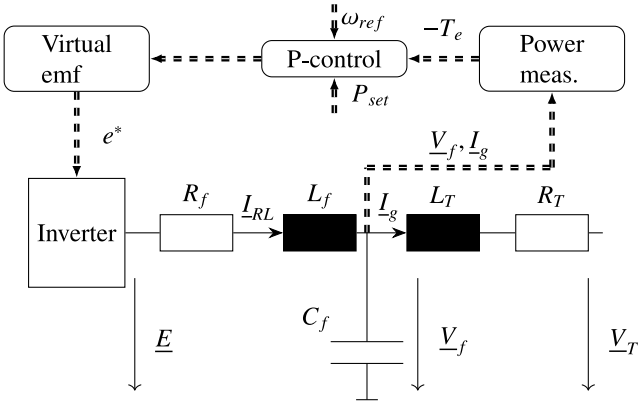

Fig. 6. Simplified inverter diagram of the synchronverter.

The active power control side is determined in the synchronverter topology, which imitates the swing equation by calculating a virtual angular frequency ω_c and a virtual angle δ_c:

$$\dot{\omega}_c = \frac{1}{J_c}(-d_c(\omega_{ref} - \omega_c) - T_e + T_m) \tag{20}$$

$$\dot{\delta}_c = \omega_c \tag{21}$$

$$\dot{e}^* = \delta_c M_f i_f \sin(\delta_c). \tag{22}$$

The inverter inertia and damping control parameters are J_c and d_c, while T_e describes the electrical torque and M_f the mutual inductance. The virtual generated voltage is termed e^*. The mechanical torque T_m is based on a power setpoint P_{set}, which is provided during normal operation. We only utilize the inertial response of the batteries, hence, $P_{set} = 0$.

The inverter parameters are presented in Table 2. We also added a frequency deadband to the inverter controller. It avoids taking actions when the deviation is too small.

Table 2
Synchronverter parameters.

Description	Symbol	Value
Filter resistance	R_f	0.375 mΩ
Filter inductance	L_f	0.3 mH
Filter capacitance	C_f	0.25 mF
Transformer resistance	R_T	0.22 mΩ
Transformer inductance	L_T	0.3 mH
Virtual Inertia	J_c	$4.052 \cdot 10^{-4}$
Virtual Damping	d_c	0.679

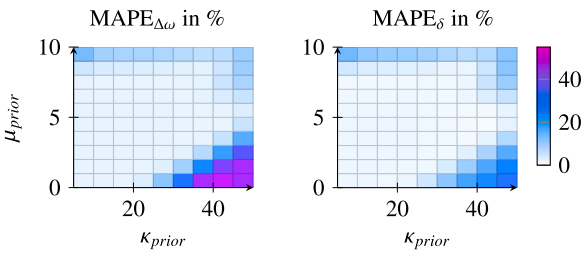

Fig. 7. MAPE sensitivity to system parameter prior $p(\lambda)$.

3.3. Synchronous generator model

The synchronous generator is represented by a third order system, that models the dynamic behavior of frequency deviation $\Delta\omega_{gen}$ and angle δ_{gen} based on the swing equation and the dynamics of governor control [18]

$$\dot{\delta}_{gen} = \Delta\omega_{gen} \tag{23}$$

$$\Delta\dot{\omega}_{gen} = \frac{1}{m_{gen,k}}(-d_{gen,k}\Delta\omega_{gen} - \sum_j B_{kj}\sin(\delta_j - \delta_k) + P_{m,k} + P_{gov,k}) \tag{24}$$

$$\dot{P}_{gov,k} = -\frac{1}{T_s}(\Delta\omega_{gen} + P_{gov,k}). \tag{25}$$

The $m_{gen,k}$ and $d_{gen,k}$ are the inertia and damping of the generator and T_s the governor time constant.

4. Results

We compare performance by assessing the mean absolute percentage error (MAPE) of the original trajectories $\boldsymbol{x}$ and the reconstruction $\hat{\boldsymbol{x}}(\hat{\lambda})$ based on the estimated system parameters $\hat{\lambda}$. The MAPE is defined as follows:

$$\text{MAPE} = 100\% \cdot \frac{1}{n_z}\sum_i^{n_z}\left|\frac{x_i - \hat{x}_i(\hat{\lambda})}{x_i}\right|. \tag{26}$$

We evaluate the sensitivity of the parameter estimation error to weakly-informative prior design in the following based on (13). The aim is to balance the amount of information in the prior and avoid non-informative priors. A totally non-informative prior would bypass the Bayesian approach, while a prior containing too much information would cause a biased estimate. In general, a similar sensitivity of the MAPEs can be found with respect to κ_{prior} and μ_{prior} for $\Delta\omega$ and δ in Fig. 7. When μ_{prior} is small, i.e. range of approximately $[0, 5]$, small estimation errors are achieved. A similar behavior is found for κ_{prior}, which constantly achieves low errors in the range of up to 25 for both quantities. Both behaviors are more pronounced for MAPE$_{\Delta\omega}$ than for MAPE$_{\delta}$. High μ_{prior} and κ_{prior} lead to less informative priors. This is indicated by the decreased density in Fig. 2. Fig. 7 reveals that too little information in the prior result in a larger error. Based on these results, both parameters seem to be best set in the medium region around $\mu_{prior} = [1, 4]$ and $\kappa_{prior} = [10, 20]$. For this paper, we use $\mu_{prior} = 1.0$ and $\kappa_{prior} = 25.0$ in all test scenarios.

4.1. Influence of epistemic uncertainty

In this section, we compare the BPINN for system identification under model uncertainty on four different systems for three dynamic settings each and contrast the performance with SINDy and PINN. The results are presented in Table 3. These three approaches, BPINN, PINN and SINDy are substantially different. SINDy uses a point-wise method to obtain the derivatives of the states $\boldsymbol{x}$ while PINN and BPINN apply automatic differentiation. The latter technique calculates the derivatives with respect to inputs $\boldsymbol{u}$ by dismantling the surrogate model, i.e. the BNN, into primitives with known derivatives. These are combined using the chain rule based on a computational graph. When the surrogate model is accurate, the derivative will also be accurate, ultimately leading to precise system parameter estimates. In addition, the BPINN provides richer information through the posterior standard deviation, which is not given by either SINDy or PINN. In this paper, we provide the reconstruction error MAPE for each state $\boldsymbol{x}$ and also give the corresponding posterior standard deviation 2σ from the BPINN.

The results demonstrate that SINDy achieves lower errors than BPINN and also PINN for the SMIB system estimation. The data utilized accurately represent the formulation of the regression problem, that is, (15), thus, SINDys point-wise approach is advantageous here. On the contrary, the BPINN is unable to achieve lower errors, since it fits a family of surrogate models. The determined distribution can only be narrowed down to a certain point with confidence. It also becomes apparent that the posterior standard deviation becomes wider with decreasing dynamics, i.e. from fast to slow. This stems from the sensitivity of the trajectories $\Delta\omega$, δ to changes in the system parameters λ. Slower dynamics tend to lead to less sensitive state trajectories $\boldsymbol{x}$ considering a change in parameters λ. This decreases the BPINNs confidence, since there are fewer contradictory values in the initial distribution. A similar behavior can be found for all test grids: the fastest dynamics always result in the most narrow distribution compared to slower dynamics. For the same reason, we find larger errors in slower dynamics for the PINN. The errors MAPE$_{\Delta\omega}$ are comparable to the MAPE$_{\delta}$ errors for SINDy in SMIB. The BPINN produces slightly larger errors for MAPE$_{\Delta\omega}$. A similar behavior is found for the PINN.

The previous paragraph evaluates the SMIB benchmark results. In the following, we introduce epistemic uncertainty with the three other grids that incorporate IBRs. The results show that the BPINN is able to achieve significantly lower errors than SINDy and also the PINN in all cases. Most times, the BPINN error is smaller by factor ten compared to SINDy error, in some cases it is even close to factor 90. This stems from the fact that SINDys estimation approach is not beneficial anymore, since the regression problem, (15), does not describe the power system behavior to the full extent. SINDy often determines a parameter set that only partially represents the system behavior. The PINN and BPINN show better performance. Both seek to obtain a surrogate model which enables state prediction for any point in time. The system parameter estimates $\hat{\lambda}$ are calculated based on these state estimates and the corresponding derivatives. This averages out underlying effects, since BPINN and PINN use automatic differentiation compared to point-wise differentiation. The BPINN achieves better performance compared to the PINN due to its Bayesian nature, which aims to exclude contradictory parameters from the posterior distribution. Higher uncertainty causes a distribution that consists of widespread parameters, thus, it most likely covers the correct solution. The BPINN achieves errors lower than PINN by factor two to three in most cases.

For all algorithms, the error decreases with slower dynamics for the majority of dynamic settings. This stems from the fact that the fast IBR dynamics vanish in the dynamics of the overall grid. In faster dynamic scenarios, the IBR dynamics are more pronounced and dominate the system behavior. This effect can be distinctively observed in the 14-bus system, where SINDy achieves high errors in fast dynamics and

Table 3
Influence of epistemic uncertainty in 3-Bus, 14-bus and 118-bus system and different dynamic scenarios on BPINN, PINN and SINDy.

System	Algorithm	Fast dynamics				Medium dynamics				Slow dynamics			
		$\text{MAPE}_{\Delta\omega}$ [%]	2σ [%]	MAPE_{δ} [%]	2σ [%]	$\text{MAPE}_{\Delta\omega}$ [%]	2σ [%]	MAPE_{δ} [%]	2σ [%]	$\text{MAPE}_{\Delta\omega}$ [%]	2σ [%]	MAPE_{δ} [%]	2σ [%]
SMIB	BPINN	1.320	7.808	0.716	11.221	1.794	12.158	1.055	16.744	2.118	15.290	1.334	20.283
	PINN	0.273		0.284		0.247		0.288		2.821		3.940	
	SINDy	0.013		0.015		0.023		0.025		0.038		0.040	
3-bus	BPINN	9.828	12.007	5.576	9.365	5.672	5.715	3.888	4.009	2.127	10.617	1.471	9.907
	PINN	18.295		12.475		9.111		6.196		5.009		4.245	
	SINDy	56.972		55.949		60.107		59.821		43.957		43.932	
14-bus	BPINN	7.785	7.651	4.682	4.312	4.077	6.465	2.933	5.314	1.457	12.816	1.033	12.063
	PINN	14.231		9.741		6.824		5.016		3.498		3.199	
	SINDy	50.509		49.955		47.961		47.853		30.723		30.726	
118-bus	BPINN	1.079	13.543	0.832	12.308	0.496	14.510	0.379	13.310	0.506	14.643	0.374	13.418
	PINN	3.699		2.545		3.972		2.692		4.487		3.026	
	SINDy	23.174		24.624		12.891		13.941		10.862		11.933	

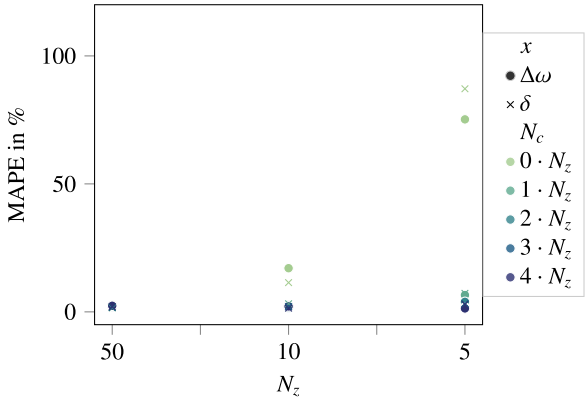

Fig. 8. Influence of collocations points (N_c) and number of samples (N_z) on the estimation accuracy (IEEE 118-bus system "fast dynamics").

significantly lower errors in slow dynamics. The 14-bus system includes a large number of IBRs compared to its size. Similar behaviors can be found for the BPINN and PINN.

Influence of sampling frequency and collocation points. Section 2 describes the BPINN and PINN ability to generate additional data points, collocation points N_c, which can potentially improve the training performance. These collocation points augment the dataset and can significantly reduce the estimation error in case of sparse data. In this paragraph, we explore the influence of collocation points N_c and the sampling frequency on the estimation accuracy. The fast dynamics scenario and the 118-bus system serve as a guiding example in Fig. 8. The results show that a small number of original samples N_z substantially increases the estimation error up to 88%. In that case, the amount of samples does not allow to create a solid trajectory and system parameter estimate. The estimation of system parameters $\hat{\lambda}$ is based on uncertain estimates $\hat{\boldsymbol{x}}$, which leads to inaccurate overall results. To address the lack of data, supplementary data points can be generated to assess the physics loss, i.e. the collocation points N_c. Fig. 8 reveals, that a small number of collocation points, $N_c = 1 \cdot N_z$, already reduces the MAPE for $N_z = 10$ close to a level comparable to $N_z = 50$. However, $N_c = 2 \cdot N_z$ is required to reach the target error. We find a similar behavior for $\text{MAPE}_{\Delta\omega}$ and MAPE_{δ}. Adding more collocation points to the training data does not significantly reduce the error when $N_z = 10$. However, a dataset of $N_z = 5$ requires $N_c = 4 \cdot N_z$ to achieve an error comparable to $N_z = 50$.

4.2. Transfer learning

The general idea behind transfer learning arises from human learning, which often uses previously learned knowledge to solve new or similar tasks in another domain. Most Machine Learning (ML) methods however assume that the training and later on operational domain are the same, which in reality is not true for most cases. This often requires a comprehensive retraining or even rebuilding of the model when the feature or domain space changes [24], resulting in manifold problems. First, it can be computationally expensive, second, the data gathering or labeling can take a long time, or even be impossible for the new domain or feature space. Therefore, it would be beneficial for the ML algorithm to reduce the need for training steps or training data from the new domain or feature space. This also holds for the BPINN.

In this section, we seek to reduce the required amount of training iterations and data through transfer learning. To achieve this, we pretrain on a SMIB with 1000 iterations and transfer the learned behavior to a larger system, the 118-bus system. The pretraining on the SMIB is used to adapt the weights and biases, so they already produce suitable estimates. The pretrained BPINN with adapted weights and biases can then be used for training with data from the actual system. The transfer learning is based on the idea that the system operator can approximate the current system behavior with a SMIB model that consists of the added dynamic constants of all active synchronous generators. This approach has been a common way to model dynamics in purely synchronous systems [25], however it is inaccurate in inverter-dominated systems. The pretraining based on the SMIB model enables the adaptation of weights and biases based on a system approximation, which enables faster training on the actual system. According to the previous introduction, our learning task will remain the same, but the learning domain changes. We strive to explore two different questions in this section: First, how many training iterations are required to achieve a comparable estimation result with pretraining compared to exclusive training in target domain? Second, can we reduce the amount of data required by pretraining on the SMIB?

Reduction of training iterations. Fig. 10 shows the evolution of MAPE over iterations for training without the pretraining. It takes around 2000 iterations to reach the final MAPE. In Fig. 9 $\text{MAPE}_{\Delta\omega}$ and MAPE_{δ} are shown over the number of iterations for training on the 118-bus system after pretraining on the SMIB. The final $\text{MAPE}_{\Delta\omega}$ is reached after 500 iterations, which is a reduction in training iterations of 75% compared to the estimations performed previously. The final MAPE_{δ} is reached after 550 iterations, which still reduces the training iterations by 72.5%. In conclusion, the results in Fig. 10 and Fig. 9 indicate that the BPINN is capable of transferring previously learned knowledge to the new space.

Reduction of data samples. This paragraph explores the transfer learning performance for sparse data. The aim is to achieve a performance comparable to full training in the target domain, i.e. Fig. 8, with pretraining and less data. Similarly to the previous analysis, we also enrich the data with collocation points. Fig. 11 demonstrates that a decrease in samples N_z results in an increased MAPE for both quantities similar to Fig. 8. Augmenting the data with $N_c = 1 \cdot N_z$ collocation points already leads to a significant improvement in the estimation accuracy for $N_z = 10$ and $N_z = 5$. However, compared to Fig. 8, it

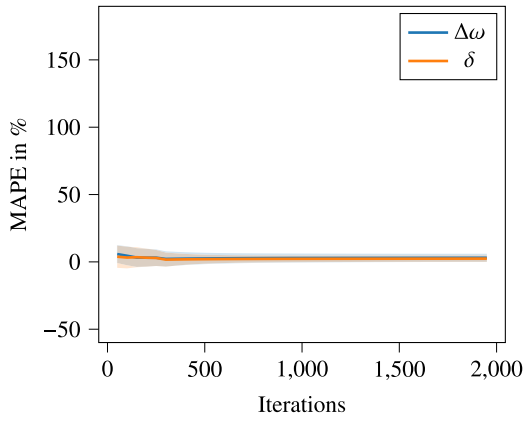

Fig. 9. Transfer learning MAPE over iterations with 2σ (2σ is represented by the colored areas, pretrained on SMIB "slow dynamics", training and estimations performed on 118-bus scenario "fast dynamics").

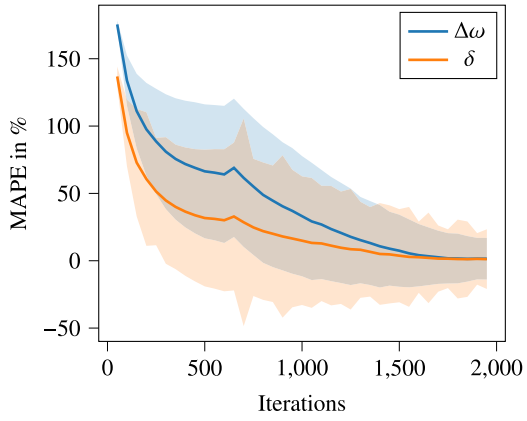

Fig. 10. Influence of iterations on estimation error MAPE with 2σ (2σ is represented by the colored areas, training and estimations performed on 118-bus scenario "fast dynamics").

can be seen that pretraining and transfer learning reduces the error, so $N_z = 10$ samples augmented with 10 collocation points already lead to the target error. Similarly, the amount of data required to reach the target error can be reduced for $N_z = 5$. After that, there are no significant improvements in the estimation error. The BPINN is able to achieve the target error with $N_z = 5$ and $N_c = 2 \cdot N_z$, which is significantly less data than in Fig. 8. We can conclude from these results that the BPINN benefits from pretraining and transfer learning. This allows us to significantly reduce the training iterations and also slightly the number of collocation points in case of sparse data.

4.3. Discussion

BPINN vs. PINN vs. SINDy. The results indicate that the BPINN and PINN cannot achieve similar results to SINDy for the SMIB system. In that case, the power system is fully represented by the regression formulation utilized (15). This advances SINDy due to its point-wise fitting approach. However, the BPINN and PINN achieve errors in the range of a few percents, which is acceptable. This reverses in the presence of IBRs, i.e. 3-bus, CIGRE 14-bus and IEEE 118-bus systems. The regression formulation (15) no longer represents the model behavior to the full extent. In this situation, it turns out to be beneficial that the BPINN fits a distribution and obtains the system parameters λ based on its learned surrogate model. The PINN also learns a surrogate model, but in contrast seeks to find a single best estimate for the system parameters λ, which appears to be detrimental in this case study.

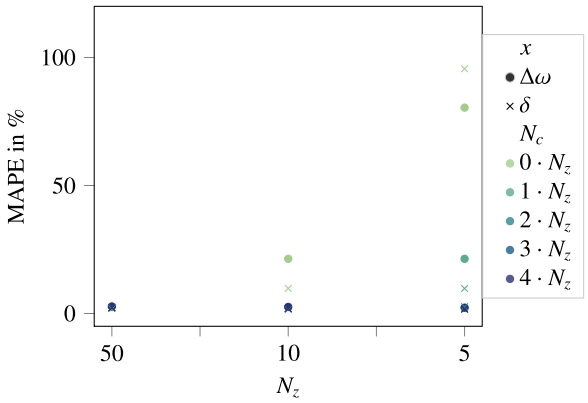

Fig. 11. Transfer learning MAPE over step size (of IEEE 118-bus data) and collocation points (trained on SMIB "slow dynamics", estimations performed on 118-bus scenario "fast dynamics").

We also aimed to compare the BPINN with the unscented Kalman filter throughout our investigations. However, this approach requires prior assumptions of process uncertainty and data noise matrices. We found these to be significantly different for various system dynamics. In a fast-changing system, such as the power system with high shares of IBRs, this would not be appropriate.

$MAPE_{\Delta\omega}$ Vs. $MAPE_{\delta}$. The results show that the BPINNs $MAPE_{\delta}$ error is often smaller than the $MAPE_{\Delta\omega}$. This stems from the shape of the individual trajectories. The $\Delta\omega$ trajectories show a more complex behavior, deviating around zero, whereas the δ trajectories move from the initial angle to the new angle. The absolute values of the individual quantities $\boldsymbol{x}$ are not influential, as they are normalized before the estimation process.

Penetration of IBRs. In the presented scenarios, the slow dynamics indicate a small share of IBRs, while the fast dynamics are dominated by IBRs. It can be seen in the results that the standard deviation of the BPINN increases with slower dynamics. The corresponding trajectories of $\boldsymbol{x}$ are less sensitive to changes in λ, leading to increased uncertainty. Similar behaviors can be found for all test grids, thus, a high share of inverters, i.e. fast dynamics, leads to higher confidence in the BPINN estimates. This is important to note, since the share of IBRs is expected to rise in future power systems.

Uncertainty quantification. The Table 3 reveals that the posterior standard deviation differs significantly for all systems in the same scenario. We expect a wider distribution in the presence of epistemic uncertainty. This expectation is based on the formulation of the quantified uncertainty (9), which depends on the epistemic uncertainty. However, in our simulation, another factor comes into play that influences the width of the posterior distribution. We previously found that slower dynamics potentially lead to increased estimation errors and consequently wider posterior standard deviation. Consequently, the uncertainty, represented by the posterior standard deviation, depends on the dynamics of the system, and the share of IBRs. Both factors also influence each other, which complicates the interpretation of the posterior standard deviation as a confidence measure. The distinction between different sources of uncertainty can only be made in theory. In practice, the BPINN would give one confidence measure and the interpretation requires significant experience.

Runtime. All experiments were performed on an Intel i7 11700 CPU. The average training time of the BPINN over all scenarios and systems was 18 s for 2000 iterations with a power system simulation step size of $T_s = 0.05$ s. Similar training times were achieved for the PINN with an average runtime of 17.8 s. SINDy training took 0.0027 s on average for one estimate. It should be noted here that the BPINN presents

its estimate as a distribution, which can be seen as computationally equivalent to performing multiple single estimates at the same time. Nevertheless, there is potential to improve the BPINN estimation speed, for example with GPU utilization.

Adaptation to real-world scenarios. The performed experiments are all based on simulation data, however, we expect the BPINN to function in real-world scenarios that may not be covered by the test scenarios for the following reasons. The BPINN always starts its estimation process from a generic setting, the weakly-informative priors, which are not focused on any previous information. These priors cover the whole range of possible parameters. Based on this, we assume that the BPINN is also able to perform estimates in the full range. In extreme cases that are difficult to estimate, the BPINN still starts the estimation process from the weakly-informative prior, therefore, it should be able to estimate extreme parameters as well. If finding an estimate is difficult, the BPINN indicates a high uncertainty. The BPINN always starts with a weakly-informative distribution and excludes unsuitable parameters, which ensures that the correct parameters remain in the distribution and if the data does not allow to reduce the distribution width, it indicates a highly uncertain estimate.

The results, i.e. Figs. 8 and 11, also showed that BPINN is able to deal with various sampling frequencies that might appear in data from real-time system monitoring. If the samples are sparse or the sampling frequency is inconsistent, BPINN can compensate for the gaps with collocation points. In doing so, BPINN is able to achieve similar estimation errors compared to high sampling rates.

In a potential real-world scenario, we assume that data can be gathered from a regular measurement gauge, such as PMU, since the sampling frequency is similar to the data from our simulation. In addition, we demonstrated in a previous study [12] that BPINN is also robust against noisy data. Based on [12] and the results presented, we assume that BPINN is also capable of working in real-world examples with data that are noisy or do not accurately represent the actual physical process.

5. Conclusion

In this paper, we explored the BPINN for system identification under model uncertainties resulting from IBRs. We evaluated the performance in four different grids: the SMIB, a 3-bus system, CIGRE 14-bus and IEEE 118-bus system equipped with multiple IBRs. The BPINN achieved lower estimation errors compared to the widely popular system identification method SINDy by a factor of 10 up to 90 in presence of IBRs and factor 2 to 3 compared to the PINN. In addition, we found that transfer learning is beneficial in BPINN training to reduce the number of iterations and the amount of required data. Pretraining on a SMIB system reduces the training time by up to 75% for estimation on the 118-bus system. The amount of required collocation points can also be reduced by pretraining and transfer learning.

CRediT authorship contribution statement

Simon Stock: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Davood Babazadeh:** Writing – review & editing, Supervision, Project administration, Investigation, Conceptualization. **Christian Becker:** Writing – review & editing, Supervision, Project administration, Conceptualization. **Spyros Chatzivasileiadis:** Writing – review & editing, Writing – original draft, Validation, Supervision, Methodology, Investigation, Formal analysis, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

References

[1] J. Zhao, A. Gómez-Expósito, M. Netto, L. Mili, A. Abur, V. Terzija, I. Kamwa, B. Pal, A.K. Singh, J. Qi, Z. Huang, A.P.S. Meliopoulos, Power system dynamic state estimation: Motivations, definitions, methodologies, and future work, IEEE Trans. Power Syst. 34 (4) (2019) 3188–3198, http://dx.doi.org/10.1109/TPWRS.2019.2894769.

[2] Y. Susuki, R. Hamasaki, A. Ishigame, Estimation of power system inertia using nonlinear koopman modes, in: 2018 IEEE Power & Energy Society General Meeting, PESGM, 2018, pp. 1–5, http://dx.doi.org/10.1109/PESGM.2018.8586007.

[3] S.L. Brunton, J.L. Procter, J.N. Kutz, Discovering governing equations from data by sparse identification of nonlinear dynamical systems, Appl. Math. 113 (15) (2016) 3932–3937, http://dx.doi.org/10.1073/pnas.1517384113.

[4] J. Stiasny, G.S. Misyris, S. Chatzivasileiadis, Physics-informed neural networks for non-linear system identification for power system dynamics, in: 2021 IEEE Madrid PowerTech, 2021, pp. 1–6, http://dx.doi.org/10.1109/PowerTech46648.2021.9495063.

[5] G.S. Misyris, A. Venzke, S. Chatzivasileiadis, Physics-informed neural networks for power systems, in: 2020 IEEE Power Energy Society General Meeting, PESGM, 2020, pp. 1–5, http://dx.doi.org/10.1109/PESGM41954.2020.9282004.

[6] M. Raissi, P. Perdikaris, G. Karniadakis, Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, J. Comput. Phys. 378 (2019) 686–707, http://dx.doi.org/10.1016/j.jcp.2018.10.045.

[7] P. Sun, R. Wu, H. Wang, G. Li, M. Khalid, G. Konstantinou, Physics-informed fully convolutional network-based power flow analysis for multi-terminal mvdc distribution systems, IEEE Trans. Power Syst. (2024) 1–13, http://dx.doi.org/10.1109/TPWRS.2024.3382266.

[8] T. Su, J. Zhao, Y. Pei, F. Ding, Probabilistic physics-informed graph convolutional network for active distribution system voltage prediction, IEEE Trans. Power Syst. 38 (6) (2023) 5969–5972, http://dx.doi.org/10.1109/TPWRS.2023.3311638.

[9] W. Wang, N. Yu, Estimate three-phase distribution line parameters with physics-informed graphical learning method, IEEE Trans. Power Syst. 37 (5) (2022) 3577–3591, http://dx.doi.org/10.1109/TPWRS.2021.3134952.

[10] N. Petra, C.G. Petra, Z. Zhang, E.M. Constantinescu, M. Anitescu, A bayesian approach for parameter estimation with uncertainty for dynamic power systems, IEEE Trans. Power Syst. 32 (4) (2017) 2735–2743, http://dx.doi.org/10.1109/TPWRS.2016.2625277.

[11] L. Yang, X. Meng, G.E. Karniadakis, B-pinns: Bayesian physics-informed neural networks for forward and inverse pde problems with noisy data, J. Comput. Phys. 425 (2021) 109913, http://dx.doi.org/10.1016/j.jcp.2020.109913.

[12] S. Stock, J. Stiasny, D. Babazadeh, C. Becker, S. Chatzivasileiadis, Bayesian physics-informed neural networks for robust system identification of power systems, in: 2023 IEEE Belgrade PowerTech, 2023, pp. 1–6, http://dx.doi.org/10.1109/PowerTech55446.2023.10202692.

[13] A. Kendall, Y. Gal, What uncertainties do we need in bayesian deep learning for computer vision?, in: I. Guyon, U.V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, R. Garnett (Eds.), Advances in Neural Information Processing Systems, Vol. 30, Curran Associates, Inc., 2017.

[14] I. Kononenko, Bayesian neural networks, Biol. Cybernet. 61 (5) (1989) 361–370, http://dx.doi.org/10.1007/BF00200801.

[15] O. Graf, P. Flores, P. Protopapas, K. Pichara, Error-aware b-pinns: Improving uncertainty quantification in bayesian physics-informed neural networks, 2022, arXiv:2212.06965.

[16] Q. Liu, D. Wang, Stein variational gradient descent: A general purpose bayesian inference algorithm, in: D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, R. Garnett (Eds.), Advances in Neural Information Processing Systems, Vol. 29, Curran Associates, Inc., 2016.

[17] N.P. Lemoine, Moving beyond noninformative priors: why and how to choose weakly informative priors in bayesian analyses, Oikos 128 (7) (2019) 912–928, http://dx.doi.org/10.1111/oik.05985.

[18] A. Bergen, D. Hill, A structure preserving model for power system stability analysis, IEEE Trans. Power Appar. Syst. PAS-100 (1) (1981) 25–35, http://dx.doi.org/10.1109/TPAS.1981.316883.

[19] A. Kaptanoglu, B. de Silva, U. Fasel, K. Kaheman, A. Goldschmidt, J. Callaham, C. Delahunt, Z. Nicolaou, K. Champion, J.-C. Loiseau, J. Kutz, S. Brunton, Pysindy: A comprehensive python package for robust sparse system identification, J. Open Source Softw. 7 (69) (2022).

[20] ENTSO-E, System Needs Study: System Dynamic and Operational Challenges, ENTSO-E, Brussels, Belgium, 2023.

[21] E. Oerum, Future System Inertia: Technical Report, Brussels, Belgium, 2018, URL https://www.entsoe.eu/Documents/Publications/SOC/Nordic/2018/System-inertia.zip.

[22] Q.-C. Zhong, G. Weiss, Synchronverters: Inverters that mimic synchronous generators, IEEE Trans. Ind. Electron. 58 (4) (2011) 1259–1267, http://dx.doi.org/10.1109/TIE.2010.2048839.

[23] R. Rosso, J. Cassoli, S. Engelken, G. Buticchi, M. Liserre, Analysis and design of lcl filter based synchronverter, in: 2017 IEEE Energy Conversion Congress and Exposition, ECCE, 2017, pp. 5587–5594, http://dx.doi.org/10.1109/ECCE.2017.8096930.

[24] S.J. Pan, Q. Yang, A survey on transfer learning, IEEE Trans. Knowl. Data Eng. 22 (10) (2010) 1345–1359, http://dx.doi.org/10.1109/TKDE.2009.191.

[25] CIGRE, Impact of High Penetration of Inverter-Based Generation on System Inertia of Networks, CIGRE, Paris, France, 2021.