IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 12 December 2023, accepted 2 January 2024, date of publication 12 January 2024, date of current version 24 January 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3353369

RESEARCH ARTICLE

Interpretable Deep Learning for Nonlinear System Identification Using Frequency Response Functions With Ensemble Uncertainty Quantification

WILL R. JACOBS, VISAKAN KADIRKAMANATHAN, (Member, IEEE), AND SEAN R. ANDERSON

Department of Automatic Control and Systems Engineering, The University of Sheffield, S1 3JD Sheffield, U.K.

Corresponding author: Sean R. Anderson (s.anderson@sheffield.ac.uk)

This work was supported by the UK's Engineering and Physical Sciences Research Council (EPSRC) Programme Grant EP/S016813/1.

ABSTRACT Deep learning methods contain powerful tools for modelling nonlinear dynamic systems. However, whilst these models are useful for predicting outputs, they tend to be described by complicated black box equations that lack interpretability. They are therefore not so useful for giving insight into system dynamics, and importantly, insight into *why* a system produces a certain output in response to a given input. This paper presents a novel method for interpreting and comparing deep learning models for nonlinear system identification, using nonlinear output frequency response functions (NOFRFs). NOFRFs describe nonlinear dynamic system behaviour in the frequency-domain using one-dimensional functions, in a manner similar to how Bode plots are used for analysing linear dynamic systems. This is a classical way of interpreting and understanding system behaviour, e.g. via resonances, and in the case of nonlinear systems, super and sub-harmonics, and energy transfer between frequencies. We also use uncertainty quantification via an ensemble bootstrap method to enhance the model interpretation, by propagating the model uncertainty estimates into the frequency-domain. The approach is demonstrated with gated recurrent unit (GRU) and long short term memory (LSTM) models - both are types of recurrent network used in deep learning that are analogous to nonlinear state space models. The results obtained from both a numerical example (a nonlinear mass spring damper system that exhibits energy transfer between frequencies) and a real-world nonlinear system (a magneto-rheological damper) show that it is possible to gain valuable insight and interpretation of the system dynamics from the NOFRFs in a way that is not possible from analysing the time-domain model equations alone.

INDEX TERMS Deep learning, nonlinear system identification, frequency response functions, uncertainty quantification, ensemble methods.

I. INTRODUCTION

The area of deep learning contains powerful methods for computational modelling of dynamic systems. However, deep learning models often lack interpretability, so although they are useful for predicting and simulating outputs they tend to be less useful for giving insight into system characteristics. The focus of this paper is to demonstrate a frequency-domain approach for interpreting deep learning models used for nonlinear system identification. Frequency-domain methods are a classical way of analysing nonlinear dynamic systems, and can give crucial insight into system behaviour including the existence of super and sub-harmonics, and energy transfer between frequencies [1]. Frequency-domain analysis of the

The associate editor coordinating the review of this manuscript and approving it for publication was Dong Shen.

type discussed here has been widely applied to different types of system including crack detection in structures [2], understanding time-varying dynamics in artificial muscle actuators [3] and condition monitoring in railway and manufacturing systems [4].

Nonlinear system identification (NSID) is concerned with data-driven modelling of nonlinear dynamic systems. There are many model classes available for NSID, such as nonlinear auto-regressive (NARX) models [5], shallow neural networks [6] and fuzzy logic [7] - a useful unifying overview of these methods is given in [8], which points out that the differences are primarily based on the choice of the basis function expansion of inputs, e.g. polynomials for NARX models, radial basis functions in shallow neural networks or first order basis splines in fuzzy logic. A further model class, nonlinear state-space models, have also been used in NSID, using particle filter methods for state estimation within the expectation-maximisation (EM) algorithm [9], [10]. Deep learning methods are now also becoming popular in nonlinear system identification, particularly using recurrent networks, which are analogous to nonlinear state-space models [11]. However, recurrent networks in NSID are usually identified using stochastic gradient descent and the backpropagation through time algorithm, which is relatively simple to apply compared to EM methods and more widely supported by popular modern software tools. For these reasons, in this paper, we consider the use of deep learning models for NSID.

Model interpretation is an important challenge across the breadth of deep learning methods because deep learning models tend to be in a complex, black-box form that are difficult to understand via model equations [12]. This is also true for the specific case of deep learning in NSID and methods for interpreting these types of model are currently lacking, which is the research gap we aim to address here. It has been noted that the very idea of model interpretation does not have a standard definition [12], [13], although it is often described as giving insight into *why* an output is predicted as opposed to just *what* is predicted [14]. In system identification, we might wish to understand why a model behaves in a certain way, for instance, whether it is due to a resonant mode of behaviour, or the bandwidth of the system dynamics and so on. These questions become even more complex for nonlinear systems where, unlike for linear systems, energy can be transferred across frequencies [1]. The frequency-domain is a natural perspective from which to analyse system dynamics and interpret model behaviour because it gives insight into and explains system behaviours in a way that would not be possible from inspecting black box model equations. Therefore, the main contribution of this paper is to demonstrate an approach to interpreting deep learning models for NSID in the frequency-domain. This would be referred to as *post hoc* analysis under certain established systems of model interpretation [12].

A nonlinear system's dynamic behaviour can be interpreted in the frequency-domain using methods such as the generalised frequency response functions (GFRFs) [15]. The GFRFs can be obtained directly from input-output data [16], [17], [18] or in a model-based framework [19], [20], [21], where the model is excited across a range of frequencies by harmonic probing [22], [23]. The GFRFs are based on the multidimensional Fourier transform of the Volterra series kernels [24] and as such are multidimensional descriptors of the nonlinear dynamics. This makes them complex to evaluate and analyse, especially because the dimensionality increases with the order of nonlinearity. The nonlinear output frequency response functions (NOFRFs) [1], [25] are an alternative to the GFRFs. The key advantage of NOFRFs over GFRFs is that the NOFRFs are one-dimensional descriptors, and give information analogous to the Bode plot for linear systems but at different orders of nonlinearity. The magnitude of each nonlinear order of NOFRF can therefore be plotted as a one-dimensional function and inspected by a human for system interpretation, to explain why certain types of behaviour arise from the system. In the area of health monitoring, NOFRFs have been combined with deep learning methods to improve the detection of faults [26], [27], [28], which is linked to the work conducted here. One of the main contributions of this paper is to develop the use of NOFRFs to analyse and interpret recurrent models used in deep learning, specifically with gated recurrent units (GRUs) [29], and long short term memory (LSTM) units [30], both of which are types of nonlinear state-space model.

Uncertainty quantification is important in NSID to assess the accuracy and trustworthiness of model predictions. Similarly, it is also important for model interpretation because it reveals where model interpretations are trustworthy, and enables more effective comparison amongst models. Uncertainty quantification for nonlinear dynamic models can be obtained using Bayesian methods such as variational inference [31] and Markov chain Monte Carlo (MCMC) [32], [33]. A number of similar methods for uncertainty quantification also exist in the deep learning literature, which can be divided into two main classes of Bayesian methods and ensemble methods [34]. Bayesian methods can use variational inference [35], [36], stochastic gradient MCMC [37], scalable weight averaging [38] and Monte Carlo dropout [39], [40]. Ensemble methods include deep ensembles based on random initialisations of the training process and shuffling data [41], sub-ensembles with weight sharing [42], and network pruning to reduce the ensemble size [43]. An appealing aspect of ensemble methods is that they tend to be simple to implement and require little tuning in terms of hyperparameters [41].

Methods exist for propagating uncertainty in NARX models into the frequency-domain using Monte Carlo sampling [44] and analytic methods [45] but this has not yet been done for deep learning models. This is an important research gap to address because it will extend the use of NOFRF analysis with uncertainty quantification to models identified by deep learning identification methods, which

are now becoming more popular. We address this research gap here by using an ensemble deep learning method for uncertainty quantification based on the bootstrap [46], [47] to characterise uncertainty in the NOFRFs. The bootstrap is a sampling-based method of statistical inference, which as with other ensemble methods is appealing for its simplicity to implement, even for complex models, which is why we use it here. The bootstrap has been widely used with time-series models, including the block-bootstrap and the residual bootstrap [48], [49], [50], [51]. In this paper we use the stationary bootstrap [52], a type of variable length block-bootstrap, to quantify uncertainty in the recurrent network models. In the stationary bootstrap, variable length blocks of time-series data are sampled with replacement from the full training data set to train an ensemble of models, and, for the first time, we propagate the uncertainty derived from the model ensemble into the frequency-domain using NOFRFs, to enhance system interpretation.

To demonstrate the approach, from system identification to model interpretation using NOFRFs, we apply it to the analysis of a synthetic nonlinear system (a nonlinear mass-spring-damper) and a real-world system (a magneto-rheological damper). The results demonstrate that the system becomes both interpretable and explainable in a way that is not possible by just examining the black box model equations.

II. METHODS

A. DATA

We assume here that a dynamic system is driven by an input $\mathbf{u}_t \in \mathbb{R}^{n_u}$, where n_u is the number of inputs, and the system produces an output $\mathbf{y}_t \in \mathbb{R}^{n_y}$, where n_y is the number of outputs. The system identification task is to identify the system dynamics from pairs of input-output data, using the dataset

$$\mathcal{D} = \{(\mathbf{u}_t, \mathbf{y}_t) : 1 \leq t \leq M\} \tag{1}$$

where M is the total number of data samples. In practice, this data set is usually split into training and validation subsets, where training data is used for parameter estimation, and validation data is used for model evaluation.

The design of the input signal, $\mathbf{u}_t$, is of particular importance in NSID because it must excite the dynamics of the system in a way that enables accurate identification of the system across varying amplitudes and frequencies [53]. For linear systems, a pseudo-random binary signal (PRBS) is often preferred because it has a frequency response that resembles white noise, which excites all frequencies in the dynamic system [54]. However, for nonlinear systems, a binary signal is not sufficient to identify amplitude-dependent nonlinearities [53]. Therefore the amplitude-modulated PRBS (APRBS) signal can be used as an alternative, which addresses this limitation [55]. The main parameters in the APRBS are the minimum and maximum amplitudes, the length of the signal, and the frequency-domain passband of the signal. The APRBS signal was used here to excite the nonlinear system under investigation.

B. MODEL DEFINITION

This section describes the linear state-space model, simple recurrent neural network (RNN), GRU and LSTM recurrent networks, giving a unifying overview of these model classes.

1) LINEAR STATE SPACE MODEL

A standard linear state-space model for dynamic systems is defined in terms of a state equation and output equation,

$$\mathbf{x}_t = A\mathbf{x}_{t-1} + B\mathbf{u}_{t-1} \tag{2}$$

$$\hat{\mathbf{y}}_t = C\mathbf{x}_t \tag{3}$$

where $\mathbf{x}_t \in \mathbb{R}^{n_x}$ is the state vector, $\hat{\mathbf{y}}_t \in \mathbb{R}^{n_y}$ is the model output, $A \in \mathbb{R}^{n_x \times n_x}$ is the state transition matrix, $B \in \mathbb{R}^{n_x \times n_u}$ is the input matrix, and $C \in \mathbb{R}^{n_y \times n_x}$ is the output matrix.

2) SIMPLE RECURRENT NEURAL NETWORK MODEL

The state equation for the simple RNN is similar to the linear state-space model but with a nonlinear, hyperbolic tangent, activation of the state dynamics

$$\mathbf{x}_t = \tanh\left(A_s\mathbf{x}_{t-1} + B_s\mathbf{u}_{t-1}\right) \tag{4}$$

where $A_s \in \mathbb{R}^{n_x \times n_x}$, $B_s \in \mathbb{R}^{n_x \times n_u}$ are learnable weight matrices and where, unlike for the linear model, we might include bias parameters as a column in the B matrix (increasing the dimensionality of the columns by one, $B_s \in \mathbb{R}^{n_x \times n_u+1}$) and augment the input with a row of ones to include this bias term.

Note that the output equation for the simple RNN, $\hat{\mathbf{y}}_t = C\mathbf{x}_t$, is the same as for the linear state-space model (for the simple RNN the C matrix could also include a column of bias parameters). Also note that the input is written here as $\mathbf{u}_{t-1}$ instead of the commonly used $\mathbf{u}_t$, which is standard for causal dynamic systems modelling and consistent with the linear state-space model.

3) LSTM MODEL

The LSTM recurrent model extends the simple RNN to include gate equations, which control signal flow through the model - the LSTM state update equation is [30],

$$\mathbf{x}_t = \mathbf{o}_t \odot \tanh \mathbf{s}_t \tag{5}$$

where $\mathbf{o}_t$ is the output gate defined below, $\odot$ is the Hadamard, or element-wise product, and $\mathbf{s}_t \in \mathbb{R}^{n_x}$ is the internal cell state,

$$\mathbf{s}_t = \mathbf{f}_t \odot \mathbf{s}_{t-1} + \mathbf{i}_t \odot \tanh\left(A_l\mathbf{x}_{t-1} + B_l\mathbf{u}_{t-1}\right) \tag{6}$$

where $A_l \in \mathbb{R}^{n_x \times n_x}$ and $B_l \in \mathbb{R}^{n_x \times n_u}$ are learnable weight matrices, $\mathbf{f}_t$ is the forget gate, $\mathbf{i}_t$ is the input gate, which with the output gate $\mathbf{o}_t$ are defined as

$$\mathbf{f}_t = \sigma\left(A_f\mathbf{x}_{t-1} + B_f\mathbf{u}_{t-1}\right) \tag{7}$$

$$\mathbf{i}_t = \sigma\left(A_i\mathbf{x}_{t-1} + B_i\mathbf{u}_{t-1}\right) \tag{8}$$

$$\mathbf{o}_t = \sigma\left(A_o\mathbf{x}_{t-1} + B_o\mathbf{u}_{t-1}\right) \tag{9}$$

where σ denotes the sigmoid activation function, hence the value of each gate is in the range 0-1, and $A_f, A_i, A_o, B_f, B_i, B_o$ are all learnable weight matrices that control the opening and closing of the gates. When the gate output is zero the gate is closed, when set to one the gate is fully open.

The key advantage of the LSTM recurrent network over simple RNNs is that the LSTM model can learn to set the gates such that when the forget gate is fully open, $\mathbf{f}_t = 1$, and the input gate is fully closed, $\mathbf{i}_t = 0$, then $\mathbf{s}_t = \mathbf{s}_{t-1}$, which holds the internal cell state un-modified over time-steps, mitigating the problem of vanishing and exploding gradients in backpropagation.

4) GRU MODEL

The GRU model uses gate equations similarly to the LSTM, but is a later advance that simplifies the model by reducing the number of gates - the GRU state update equation is [29],

$$\mathbf{x}_t = \mathbf{z}_t \odot \mathbf{x}_{t-1} + (1 - \mathbf{z}_t) \odot \tanh\left(A_g\left(\mathbf{r}_t \odot \mathbf{x}_{t-1}\right) + B_g \mathbf{u}_{t-1}\right) \tag{10}$$

where $A_g \in \mathbb{R}^{n_x \times n_x}$, $B_g \in \mathbb{R}^{n_x \times n_u}$ are learnable weight matrices. Note that the output equation for the GRU, $\hat{\mathbf{y}}_t = C\mathbf{x}_t$, is once again the same as for the linear state-space model (with the possible addition of bias parameters).

The GRU contains two gates: the update gate, $\mathbf{z}_t$, and the reset gate, $\mathbf{r}_t$, defined as

$$\mathbf{z}_t = \sigma\left(A_z \mathbf{x}_{t-1} + B_z \mathbf{u}_{t-1}\right) \tag{11}$$

$$\mathbf{r}_t = \sigma\left(A_r \mathbf{x}_{t-1} + B_r \mathbf{u}_{t-1}\right) \tag{12}$$

where σ denotes the sigmoid activation function, hence the value of each gate is in the range 0-1, and A_z, A_r, B_z, B_r are all learnable weight matrices that control the opening and closing of the gates. When the gate output is zero the gate is closed, and when set to one the gate is fully open. Note that the GRU model contains the simple RNN as a special case for $\mathbf{z}_t = 0$ and $\mathbf{r}_t = 1$.

The key advantage of the GRU layer over simple RNNs is similar to the LSTM in that that it can learn to set the gates so that the state is held un-modified over time-steps, e.g. when the update gate is fully open, $\mathbf{z}_t = 1$, then $\mathbf{x}_t = \mathbf{x}_{t-1}$. This enables learning of long term dependencies and mitigates the problem of vanishing and exploding gradients in backpropagation.

C. MODEL IDENTIFICATION

The section describes the identification procedure used here for the recurrent models, which consists of model structure detection, parameter estimation and model validation.

1) STRUCTURE DETECTION

Structure detection is an important problem in NSID and there are many methods that are suited to different types of model structure. For example, least squares methods [56], [57], evolutionary algorithms [58] and Bayesian methods [31], [32] have been used for NARX models. Sequential estimation methods have also been used for radial basis function neural networks [59], [60]. Hyperparameter optimization in neural networks addresses elements of the same problem as structure detection, where the number of hidden units and the number of layers can be regarded as hyperparameters. Existing methods include grid search [61], random search [62], [63], Bayesian optimization [64], [65] and evolutionary search [66].

More recently, neural architecture search (NAS) [67] has become prominent in deep learning, which could be used to address the structure detection problem in NSID. It is distinct from general hyperparameter optimization because it focuses specifically on the network architecture. Some NAS methods based on evolutionary search and reinforcement learning can be very computationally intensive because they require training many deep learning models but one-shot methods are more efficient and use one-stage training with weight sharing to improve computational efficiency [67].

In this paper, as it is primarily focused on model interpretation using frequency-domain analysis, we used a simple grid search to select the number of hidden units in the recurrent layer, i.e. the state dimension n_x.

2) PARAMETER ESTIMATION

The parameters of the recurrent model were estimated here by minimising the mean squared error loss function with L2-norm regularization (to avoid overfitting) on a training data subset of the full data, $\mathcal{D}_T \subset \mathcal{D}$,

$$J(\boldsymbol{\theta}) = \frac{1}{M_T} \sum_{t=1}^{M_T} \left\| \mathbf{y}_t - \hat{\mathbf{y}}_t \right\|_2^2 + \alpha \left\| \boldsymbol{\theta} \right\|_2^2 \tag{13}$$

where M_T is the number of training data samples and $\boldsymbol{\theta}$ comprises all the unknown parameters in the weight matrices in the recurrent network model.

The estimation algorithm used in this paper was Adam (adaptive moments) algorithm [68], a variation of stochastic gradient descent, which combines a momentum-like term, v_j, with an adaptive learning rate, r_j, and bias corrected versions of these terms, $\hat{v}_j$ and $\hat{r}_j$, where the j-th parameter update is

$$\theta_j \leftarrow \theta_j - \frac{\epsilon}{\sqrt{\hat{r}_j} + \delta} \hat{v}_j \tag{14}$$

where δ is a small offset term designed to avoid division by zero and

$$\hat{r}_j = \frac{r_j}{1 - \beta_1^t} \tag{15}$$

$$\hat{v}_j = \frac{v_j}{1 - \beta_2^t} \tag{16}$$

$$v_j \leftarrow \beta_1 v_j + (1 - \beta_1)\, g_j \tag{17}$$

$$r_j \leftarrow \beta_2 r_j + (1 - \beta_2)\, g_j^2 \tag{18}$$

and g_j is the stochastic estimate of the loss function gradient for parameter j,

$$g_j = \nabla_{\boldsymbol{\theta}} \hat{J}(\theta_j) \tag{19}$$

TABLE 1. Parameter estimation options used for the Adam algorithm.

Option	Symbol	Value
Maximum number of epochs	-	200
Mini-batch size	-	128
L2-norm regularization weight	α	1×10^{-4}
Learning rate	ϵ	0.002
Gradient decay factor	β_1	0.9
Squared gradient decay factor	β_2	0.999
Offset	δ	1×10^{-8}

where the estimate of the gradient of the loss function $\nabla_{\boldsymbol{\theta}} \hat{J}(\boldsymbol{\theta})$ was obtained here from a mini-batch of data via backpropagation through time [69].

The hyperparameters for the Adam algorithm were initially set to those used in [68] and then the learning rate ϵ was manually adjusted from 0.001 to 0.002 to give more rapid convergence. Other hyperparameters including the mini-batch size, L2-norm regularisation weight, α, and number of training epochs were manually tuned through a preliminary investigation. The parameter estimation options used here are shown in Table 1.

3) MODEL VALIDATION

The model selection procedure was validated through analysis on independent validation data, $\mathcal{D}_V \subset \mathcal{D}$ (where $\mathcal{D}_T \cap \mathcal{D}_V = \varnothing$). This was done by processing the residual errors, e_j, from predictions on the validation data where

$$e_j = \hat{y}_j - y_j \tag{20}$$

The residual errors were then used to obtain the R^2, or variance-accounted-for (VAF) metric, because it is a normalised measure of goodness-of-fit [70],

$$R^2 = 1 - \frac{\sum_{j=1}^{M_V} e_j^2}{\sum_{j=1}^{M_V} \left(\hat{y}_j - \bar{y}\right)^2}, \tag{21}$$

where M_V is the number of samples in the validation data set and $\bar{y}$ is the mean of the output data. An R^2 value of 1 indicates a perfect model fit, a value of 0 indicates a fit equivalent to the mean of the output data, and the value becomes negative for poor fits.

D. UNCERTAINTY QUANTIFICATION USING THE BOOTSTRAP

This section describes the method of uncertainty quantification used here, which is based on the bootstrap.

In ordinary bootstrapping, samples of estimation data are drawn at random with replacement from the original data set. However, the ordinary bootstrap cannot be used for dynamic models because samples are drawn independently assuming no dependence with each other, which is not the case for dynamic models, where there is correlation across time-steps. Therefore, alternatives, such as the block bootstrap have been developed for this case [71], where blocks of contiguous samples are drawn from the original data set with replacement. The method used here is based on the stationary bootstrap [52], where the length of the block is randomly selected along with the start sample, which ensures that the blocks are stationary (if the original time-series is stationary) and avoids the problem of selecting a single block length.

In the stationary bootstrap, the i-th training data set, $\mathcal{D}_i$, is selected as a block of contiguous input-output pairs of data, of randomly chosen block length, starting from a randomly chosen time-step,

$$\mathcal{D}_i = \{(\mathbf{u}_{t_i}, \mathbf{y}_{t_i}), (\mathbf{u}_{t_i+1}, \mathbf{y}_{t_i+1}), \ldots, (\mathbf{u}_{t_i+l_i-1}, \mathbf{y}_{t_i+l_i-1})\} \quad \text{for} \quad i = 1, \ldots, N_b \tag{22}$$

where t_i is the starting sample for the block, l_i is the length of the block and N_b is the number of bootstraps. The starting sample of a block is drawn from a uniform distribution

$$P(t_i = k) = \frac{1}{M} \quad \text{for} \quad k = 1, \ldots, M \tag{23}$$

where M is the length of the training data set. The block length is drawn at random from a geometric distribution

$$P(l_i = k) = (1-p)^{k-1} p \quad \text{for} \quad k = 1, 2, 3, \ldots \tag{24}$$

where $1/p$ is the mean value of the distribution. The value of p was chosen as $p = 0.001$, which ensured that on average 1000 samples were used for each nominal bootstrap training data set. The procedure was adjusted so that data sets with fewer than 500 samples were discarded (because training deep learning models requires a sufficient number of samples to avoid over-fitting), or more samples than in the training set, $\mathcal{D}_T$, were also discarded.

To implement the bootstrap, data sets $\mathcal{D}_i$ were sampled with replacement from the full training data set $\mathcal{D}_T$ and used to train N_b distinct LSTM and GRU recurrent models f_i for $i = 1, \ldots, N_b$. Training on the bootstrap data sets took place in the usual way as described above in Parameter Estimation using mini-batches of data. The number of bootstrap replicates, N_b, was chosen here to be 100. In practice, this meant training 100 models, but these models had a single hidden layer with relatively low state dimension, $1 \leq n_x \leq 100$ that took on the order of 10-20 seconds each to train, so in total this resulted in less than 30 minutes of training time (on an Intel Core i7@3.2 GHz with 6 cores and 16 GB RAM, with no GPU).

E. INTERPRETING MODEL BEHAVIOUR USING NONLINEAR OUTPUT FREQUENCY RESPONSE FUNCTIONS

This section describes the method used for interpreting deep learning models for NSID in the frequency-domain via NOFRFs. This method obtains the NOFRFs via time-domain simulation of the recurrent network and is particularly simple to implement.

1) NOFRF DEFINITION

A single output, y_t, of a homogenous nonlinear system can be defined in the frequency-domain as the sum of N nonlinear

orders of frequency response $Y_n(j\omega)$ [1],

$$Y(j\omega) = \sum_{n=1}^{N} Y_n(j\omega) = \sum_{n=1}^{N} G_n(j\omega)U_n(j\omega) \tag{25}$$

where j is the imaginary unit, ω is frequency in radians per second, and $G_n(j\omega)$ is the n-th order NOFRF defined as

$$G_n(j\omega) = \frac{\int_{-\infty}^{\infty}\cdots\int_{-\infty}^{\infty} H_n(j\omega_1,\ldots,j\omega_n)\prod_{i=1}^{n} U(j\omega_i)\mathrm{d}\omega_1\ldots\mathrm{d}\omega_n}{\int_{-\infty}^{\infty}\cdots\int_{-\infty}^{\infty}\prod_{i=1}^{n} U(j\omega_i)\mathrm{d}\omega_1\ldots\mathrm{d}\omega_n} \tag{26}$$

where H_n is the n-th order GFRF [19],

$$H_n(j\omega_1,\ldots,j\omega_n) = \int_{-\infty}^{\infty}\cdots\int_{-\infty}^{\infty} h_n(\tau_1,\ldots,\tau_n)\exp^{-j(\omega_1\tau_1,\ldots,\omega_n\tau_n)}\mathrm{d}\tau_1\ldots\mathrm{d}\tau_n \tag{27}$$

where h_n is the n-th order impulse response of the system, or Volterra kernel, τ indexes over time, and $U_n(j\omega) \neq 0$ is an input signal defined as

$$U_n(j\omega) = \frac{n^{-1/2}}{2\pi^{n-1}}\int_{-\infty}^{\infty}\cdots\int_{-\infty}^{\infty}\prod_{i=1}^{n} U(j\omega_i)\mathrm{d}\omega_1\ldots\mathrm{d}\omega_n \tag{28}$$

Note that the NOFRF definition is dependent on a specific input U_n and therefore the NOFRF is only defined for this input and will change if the input signal is changed. The consequence of this is that the system must be analysed for specific input signals of interest. This is pragmatic because systems will usually be operated with input characteristics that are known *a priori*.

2) NOFRF ESTIMATION

The NOFRFs, $G_n(j\omega)$, can be estimated using a data-driven approach [1], by probing an identified recurrent network model with a specific input signal of interest. The procedure is extended here to include uncertainty quantification derived from the bootstrap method:

1) Define the input probing signals,

$$u_t^{(k)} = \alpha_k u_t^* \quad \text{for } k = 1,\ldots,K \tag{29}$$

where a single waveform, u_t^*, is designed to excite some specific frequency band of interest, and is scaled by increasing amplitudes defined by α_k, where $\alpha_K > \alpha_{K-1} > \ldots > \alpha_1 > 0$, where K is the number of probing input signals chosen as $K \geq N$.

2) Simulate each separate recurrent model identified through bootstrap estimation K times, with the inputs $\alpha_k u_t^*$, to produce the outputs

$$\hat{y}_t^{(k,i)} = f_i\left(\mathbf{x}_t, u_t^{(k)}\right) \quad \text{for } k = 1,\ldots,K \text{ and } i = 1,\ldots,N_b \tag{30}$$

where f_i is a specific instance of recurrent model identified from bootstrapping and N_b is the number of bootstrap models.

3) Obtain the input-output frequency spectra by using the fast Fourier transform (FFT),

$$Y_{k,i}^*(j\omega) = \mathrm{FFT}\left(\hat{y}_t^{(k,i)}\right) \quad \text{for } k = 1,\ldots,K \text{ and } i = 1,\ldots,N_b \tag{31}$$

$$U_k^*(j\omega) = \mathrm{FFT}\left(\alpha_k u_t^*\right) \quad \text{for } k = 1,\ldots,K \tag{32}$$

where $Y_k^*(j\omega)$ is the output spectrum and $U_k^*(j\omega)$ is the input spectrum of the input-output signals respectively,

4) Construct a regression problem that can be solved in closed form for the NOFRFs, $G_n(j\omega)$, for each frequency ω,

$$\hat{\mathbf{G}}_{\omega,i} = \left(\mathbf{U}_\omega^{\mathrm{H}}\mathbf{U}_\omega\right)^{-1}\mathbf{U}_\omega^{\mathrm{H}}\mathbf{Y}_{\omega,i} \quad \text{for } i = 1,\ldots,N_b \tag{33}$$

where $\mathbf{U}_\omega^{\mathrm{H}}$ denotes the conjugate transpose of $\mathbf{U}_\omega$ and

$$\mathbf{Y}_{\omega,i} = \mathbf{U}_\omega\mathbf{G}_{\omega,i} \tag{34}$$

$$\mathbf{G}_{\omega,i} = \left[G_{1,i}(j\omega),\ldots,G_{N,i}(j\omega)\right]^T \tag{35}$$

$$\mathbf{Y}_{\omega,i} = \left[Y_{1,i}^*(j\omega),\ldots,Y_{K,i}^*(j\omega)\right]^T \tag{36}$$

$$\mathbf{U}_\omega = \begin{bmatrix} \alpha_1 U_1^*(j\omega) & \ldots & \alpha_1^N U_N^*(j\omega) \\ \vdots & & \vdots \\ \alpha_K U_1^*(j\omega) & \ldots & \alpha_K^N U_N^*(j\omega) \end{bmatrix} \tag{37}$$

The main tuning parameter of the NOFRF estimation procedure is the maximum nonlinear order, N. N should be chosen such that there is negligible power at subsequent orders.

Regarding design of the input probing signals, $u_t^{(k)}$, note that the input signal for each model simulation, $u_t^{(k)} = \alpha_k u_t^*$, has the same waveform u_t^*, with different amplitude scaling defined by α_k. The base probing input signal u_t^* can be designed according to user needs but a convenient form used here is the following,

$$u_t^* = \frac{3}{2\pi}\frac{\sin(2\times b\times\pi\times t) - \sin(2\times a\times\pi\times t)}{t} \tag{38}$$

where a and b define the lower and upper range of the frequency excitation in Hz, and the amplitude spectrum is approximately flat in this range.

3) NOFRF INTERPRETATION

Regarding model interpretation, a key point to note is that the NOFRFs, $G_n(j\omega)$, obtained from (33) can be analysed to give insight into the system dynamics, particularly using the magnitude spectrum given by $|G_n(j\omega)|$. The magnitude $|G_n(j\omega)|$ can be graphically analysed similarly to the magnitude spectrum in a linear system Bode plot, and for the nonlinear system can reveal behaviour such as resonances occurring at different orders of nonlinearity n, and super/sub-harmonics, as well as energy transfer across frequencies.

To interpret the model behaviour it is important to understand how the nonlinear system can generate power at

new frequency locations, which can be due to a combination of the following effects:

- The nonlinear composition of the inputs $U_n(j\omega)$, which generally contain a richer set of components than the single input spectrum $U(j\omega)$.
- The filtering effect of the NOFRF, $G_n(j\omega)$, which determines the contribution of each nonlinear order of the system to the output frequency response.
- The inter-kernel interference between the different orders of the nonlinear system, which arises from how the output spectra, $Y_n(j\omega)$, are combined in (25), and determines whether a system output is generated outside the input frequency band.

So, to summarise, there are three distinct ways the nonlinear system can affect the output spectrum: via the inputs, $U_n(j\omega)$, the filtering effect of the NOFRF, $G_n(j\omega)$, and inter-kernel interference between NOFRFs. The interplay between these effects ultimately determines whether energy will be transferred to a certain frequency. An inspection of the magnitude plot of the inputs $|U_n(j\omega)|$, NOFRFs $|G_n(j\omega)|$ and outputs $|Y_n(j\omega)|$ is therefore essential to gain insight into these phenomena and reveal the mechanisms by which energy is transmitted to specific frequencies. The analysis can be performed in much the same way as the classical analysis of linear systems using Bode plots [25]. Such an analysis has real-world applications in, for example, condition monitoring using audio signals [4] and damage detection in vibration signals [72].

The full procedure for identifying the model, and analysing and interpreting the model using NOFRFs is given in Fig. 1.

III. RESULTS

In this section, we report the results of applying the frequency-domain model interpretation on both a synthetic example, a nonlinear mass spring damper [1], [73], which exhibits energy transfer between frequencies, and a real world nonlinear system, a magneto-rheological damper [74], which demonstrates the applicability of the approach to real-world systems.

A. NONLINEAR MASS SPRING DAMPER

The NOFRF method for interpreting deep learning models is demonstrated in this section on a synthetic example, a nonlinear mass-spring-damper (MSD) used in previous studies of nonlinear systems analysis with frequency response functions [1], [73] described by

$$m\frac{d^2x(t)}{dt^2} + c\frac{dx(t)}{dt} + k_1x(t) + k_2x(t)^2 + k_3x(t)^3 = u(t) \tag{39}$$

$$y(t) = x(t) + v(t) \tag{40}$$

where m is the mass, c is the damper coefficient and k_1, k_2 and k_3 are the spring coefficients; $x(t)$ is displacement (the output variable) at time t and $u(t)$ is force (the input variable); $v(t) \sim N(0, \sigma_v^2)$ is zero mean Gaussian noise signal added to the output variable, displacement, to give the measured displacement signal $y(t)$; the noise variance was tuned to give a signal-to-noise ratio (SNR) of 20 dB. Parameter values were set to $m = 1$, $c = 20$, $k_1 = 10^4$, $k_2 = 10^7$ and $k_3 = 5 \times 10^9$, as used in previous studies [1], [73].

An advantage of studying this nonlinear system is that the GFRFs can be computed analytically for the purposes of validation. Expressions for the system's first and second order GFRFs, H_1 and H_2, were obtained here using the probing method [56], [73],

$$H_1(\omega) = \frac{1}{-m\omega^2 + cj\omega + k} \tag{41}$$

$$H_2(\omega_1, \omega_2) = -\frac{k_2}{2}H_1(\omega_1)H_1(\omega_2)H_1(\omega_1 + \omega_2) \tag{42}$$

To generate system identification data for this MSD system, an input excitation signal was designed using an APRBS (with amplitude range from -1 to 1, and passband from 0 to 50 Hz) and the system was simulated with this input using a 4th order Runge-Kutta method for 20 seconds to produce an output signal. The signals were sampled at 200 Hz, a sample time of 0.005 seconds, giving a total of 4000 samples (Fig. 2(a)-(b)). The input-output data were both normalised by their respective peak absolute values before the application of system identification methods. The dataset was split in the ratio 75:25 (15 seconds to 5 seconds) respectively for training and validation.

The MSD system was identified using a single-layer GRU network and a single layer LSTM network. A grid search was performed on the GRU model only to select the number of hidden units in the models, with n_x approximately log-spaced from 1 to 100, i.e. $n_x = 1, 3, 10, 30, 100$, with corresponding R^2 values found of 0.34, 0.42, 0.52, 0.97, 0.97 on validation data. Therefore a good accuracy-complexity trade-off was found to be $n_x = 30$ (and the LSTM model had an $R^2 = 0.95$ at this model order). The simulations of the models against validation data also demonstrated the good accuracy of the GRU and LSTM models (Fig. 2(c)).

The GRU and LSTM models as well as the true system defined in (39) were then probed by an input signal as defined in (38), with a flat amplitude spectrum between the limits $a = 30$ Hz and $b = 50$ Hz and zero otherwise (Fig. 2(d)-(e)). The output amplitude spectrum of the true system was compared to the GRU and LSTM models using an FFT. The magnitude of the output frequency response of the data falls outside of the 95% confidence limits but despite this, it can be seen that the main characteristics of the output frequency response are captured by the models (Fig. 2(f)). It is noticeable that the uncertainty is higher at lower frequencies - a point that is not appreciable from the time-domain simulations in Fig. 2(c). A striking feature of the output amplitude spectrum in Fig. 2(f) at around 15 Hz is the resonant peak, which results from energy transfer (which we know because the system is not excited in this range). The question for model interpretation is - *why* does this resonant peak occur here?

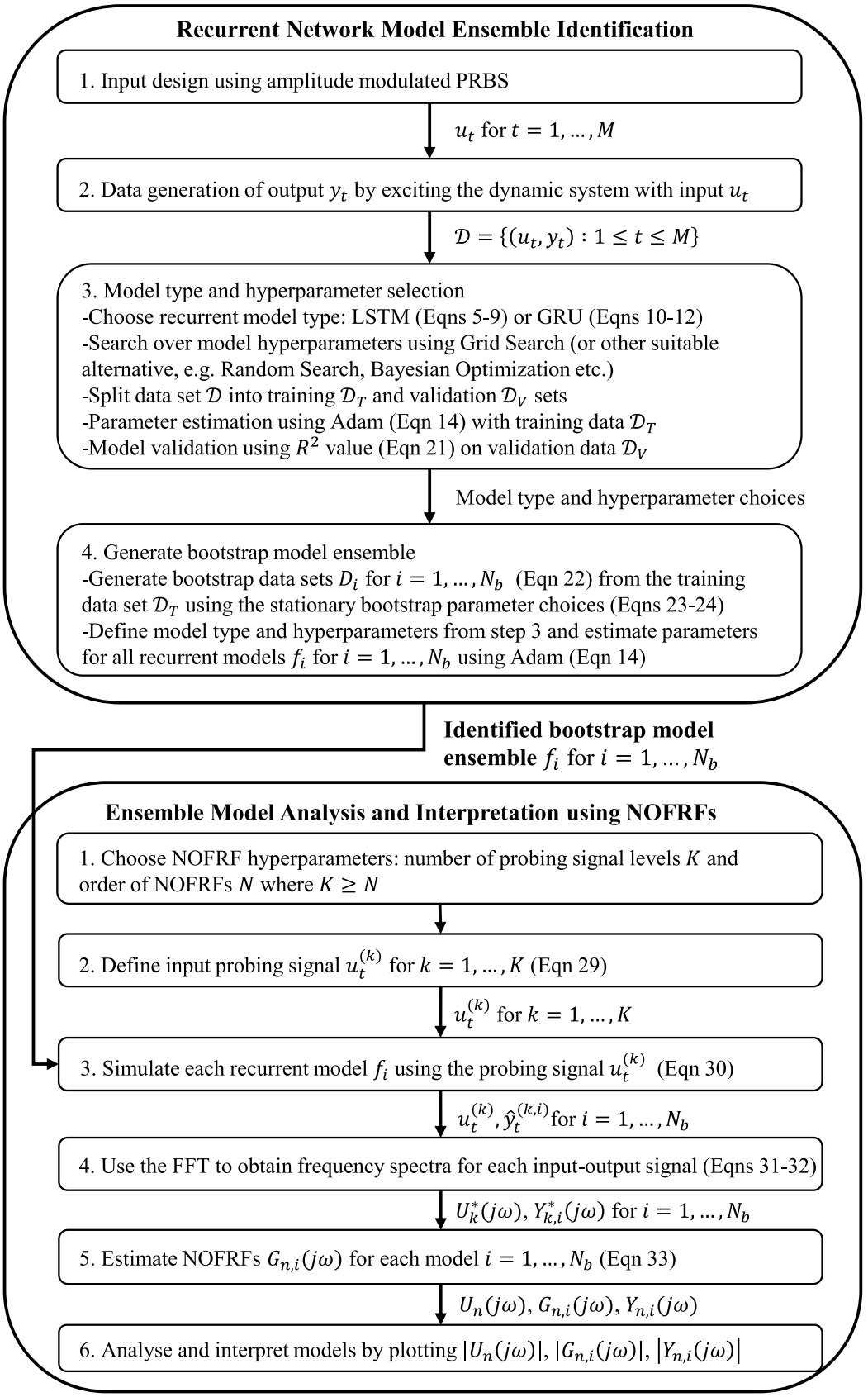

FIGURE 1. Model ensemble identification, and NOFRF analysis and interpretation procedure.

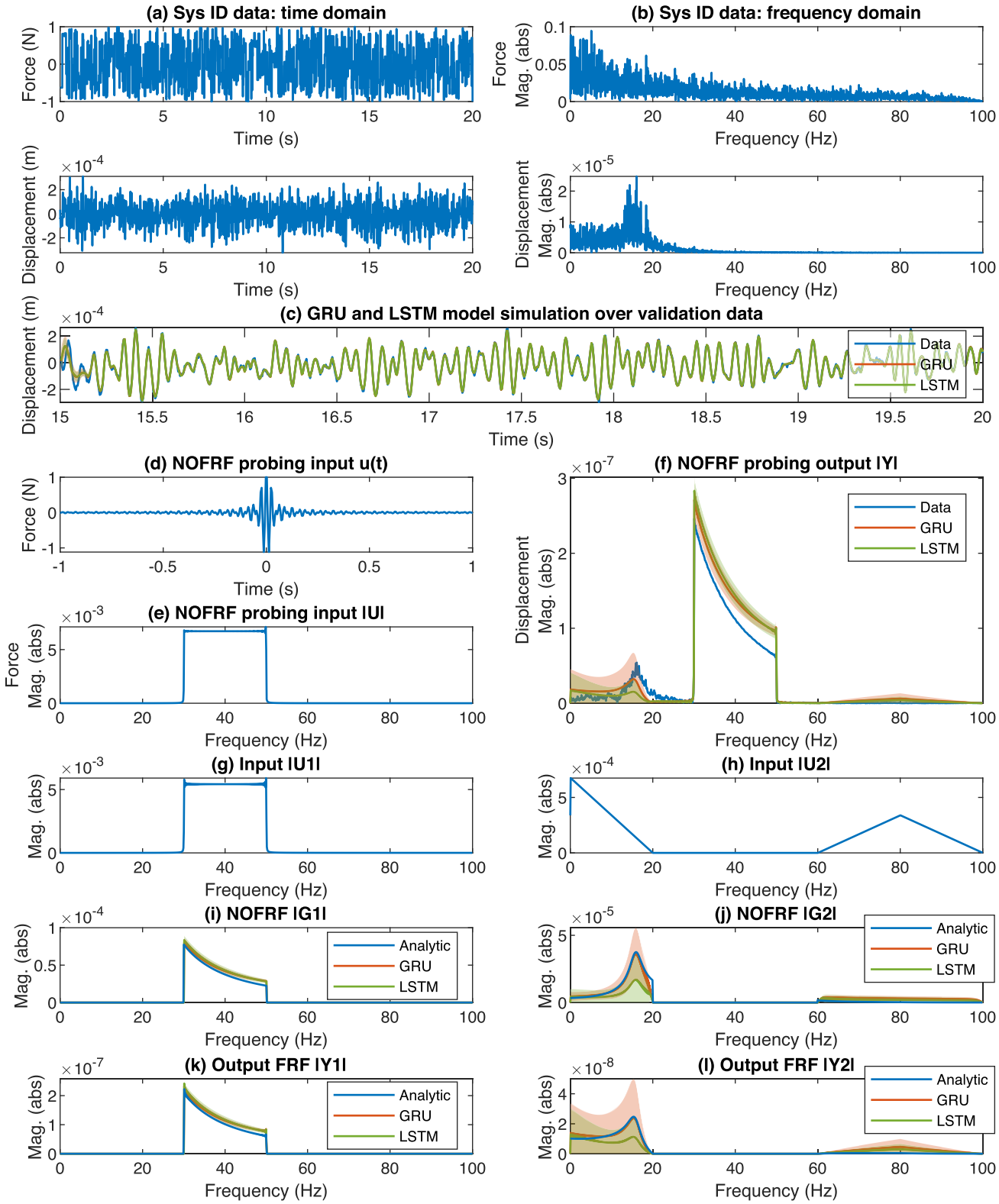

FIGURE 2. Mass spring damper modelling and interpretation using frequency response functions. (a)-(b) System identification input-output data in the time and frequency-domains. (c) GRU and LSTM model predictions compared to validation data. (d)-(e) NOFRF input data used to probe the nonlinear system. (f) NOFRF probing output in the frequency-domain. (g)-(h) First and second order inputs. (i)-(j) First and second order NOFRFs. (k)-(l) First and second order output frequency responses. In each plot, a shaded region indicates the 95% confidence interval derived from uncertainty quantification using bootstrapping.

To answer *why* the system exhibits these dynamics and interpret the GRU and LSTM models, we examine the first and second-order input FRFs, NOFRFs and output frequency responses in Fig. 2(g)-(l): we can see that as expected the first order FRFs (Fig. 2(g), (i), (k)) exhibit behaviour one would associate with a linear system, with no energy transfer or outputs outside of the excited frequency range (30-50 Hz). In contrast, the second order input FRF has power in the range 0-20 Hz (Fig. 2(g)), the second order NOFRF (Fig. 2(j)) has a resonant peak around 15 Hz, and the second order output frequency responses (Fig. 2(k)) exhibits this resonant peak as a result of the input and the filtering effect of the NOFRF. This, therefore, explains why the overall output in Fig. 2(f) contains this resonant mode, and thus we can explain and interpret the model behaviour.

We can also validate the NOFRF analysis from the analytic solution obtained from the GFRF equations defined in (41) and (42) - notice that there is good agreement between the analytic solution and the NOFRFs derived from the identified GRU and LSTM models (Fig. 2(i)-(l)).

It is also notable that the lower frequency contents (below 20 Hz) have larger uncertainties - this may be due to the fact that in finite data records, there will always be less low-frequency data than high-frequency, for example, in 20 seconds of data, there are effectively 20 (non-overlapping) examples of 1 Hz waves, but 200 examples of 10 Hz waves, so when bootstrapping, the model variability will be much higher at the lower frequencies because there are fewer examples of low-frequency behaviour. An additional interpretation we might draw from this is that the model behaviour at low frequency will be more uncertain and therefore might be more prone to drift over time.

Finally, it is worth noting that although the GRU and LSTM model equations are completely different, the NOFRF analysis gives close agreement between the two models and therefore provides a unifying insight into the system dynamics, emphasising that irrespective of the form of black box model equations, this interpretation procedure can be consistently applied across deep learning model descriptions.

B. MAGNETO-RHEOLOGICAL DAMPER

The NOFRF method for interpreting deep learning models is demonstrated in this section on a real world system, a magneto-rheological (MR) damper using measurements of its velocity (input) and the damping force (output) sampled at 200 Hz. The experimental data is described in Wang et al. [74] and is obtained here as a standard dataset provided in the Mathworks Matlab System Identification Toolbox [75].

MR dampers typically consist of magnetically polarizable particles dispersed in a fluid such as oil. The viscosity of this fluid can be altered by the application of an magnetic field acting across the magnetic particles. Therefore, MR dampers can be used to actively control damping force by manipulating the viscosity via an electromagnet, which in turn is controlled by voltages/current. In this dataset, the MR damper was fixed to the ground at one end and connected at the other end to a shaker table generating vibrations. The input-output data was sampled every 0.005 s, giving a total of 3499 samples. The input-output data were both normalised by their respective peak absolute values before the application of system identification methods. The dataset was split in the ratio 75:25 for training and validation sets respectively.

The MR system was identified using a single-layer LSTM network and a single layer GRU network. A grid search was performed to select the number of hidden units in the GRU and LSTM models, with n_x approximately log-spaced from 1 to 100, i.e. $n_x = 1, 3, 10, 30, 100$, with corresponding R^2 values found to be $-0.38, 0.72, 0.93, 0.96, 0.98$. Therefore, a good accuracy-complexity trade-off was found to be $n_x = 10$ (and the LSTM model had an $R^2 = 0.94$ at this model order). The simulations over validation data also demonstrated the good accuracy of each model (see Fig. 3(c)).

The GRU and LSTM models were then probed by an input signal as defined in (38), with a flat amplitude spectrum between the limits $a = 8$ Hz and $b = 10$ Hz (Fig. 3(d)-(e)). In this case, the probing output of the models could not be compared to the true system, which was not available to experiment on with custom inputs. However, the output amplitude spectra of the GRU and LSTM models were compared to each other using an FFT, which demonstrated good agreement, particularly within the 95% confidence limits (Fig. 3(f)). A striking feature of the output amplitude spectrum in Fig. 3(f) is that there is power in the response at low frequency (<5 Hz), which must result from energy transfer because we know because the system is not excited in this range. Once again, the question for model interpretation is - *why* does the system exhibit this effect?

To answer *why* the system exhibits these dynamics and interpret the GRU and LSTM models, we examine the first and second order input FRFs, NOFRFs and output frequency responses in Fig. 3(g)-(l): the first order FRFs (Fig. 3(g), (i), (k)) exhibit standard behaviour with no energy transfer or outputs outside of the excited frequency range (8-10 Hz). In contrast, the second order input FRF has power in the range 0-2 Hz (Fig. 3(g)), the second order NOFRF (Fig. 3(j)) also has power in this range, and the second order output frequency responses (Fig. 3(k)) therefore exhibit power in this range as a result of the input and the filtering effect of the NOFRF. This, therefore, enables us to interpret the model behaviour and explain why the overall output in Fig. 3(f) contains this low frequency response.

IV. DISCUSSION

The aim of this paper was to develop a model interpretation approach for deep learning models used in nonlinear system identification. We proposed a frequency-domain approach because this gives insight into why a system generates particular dynamic behaviour. Specifically, we used NOFRFs

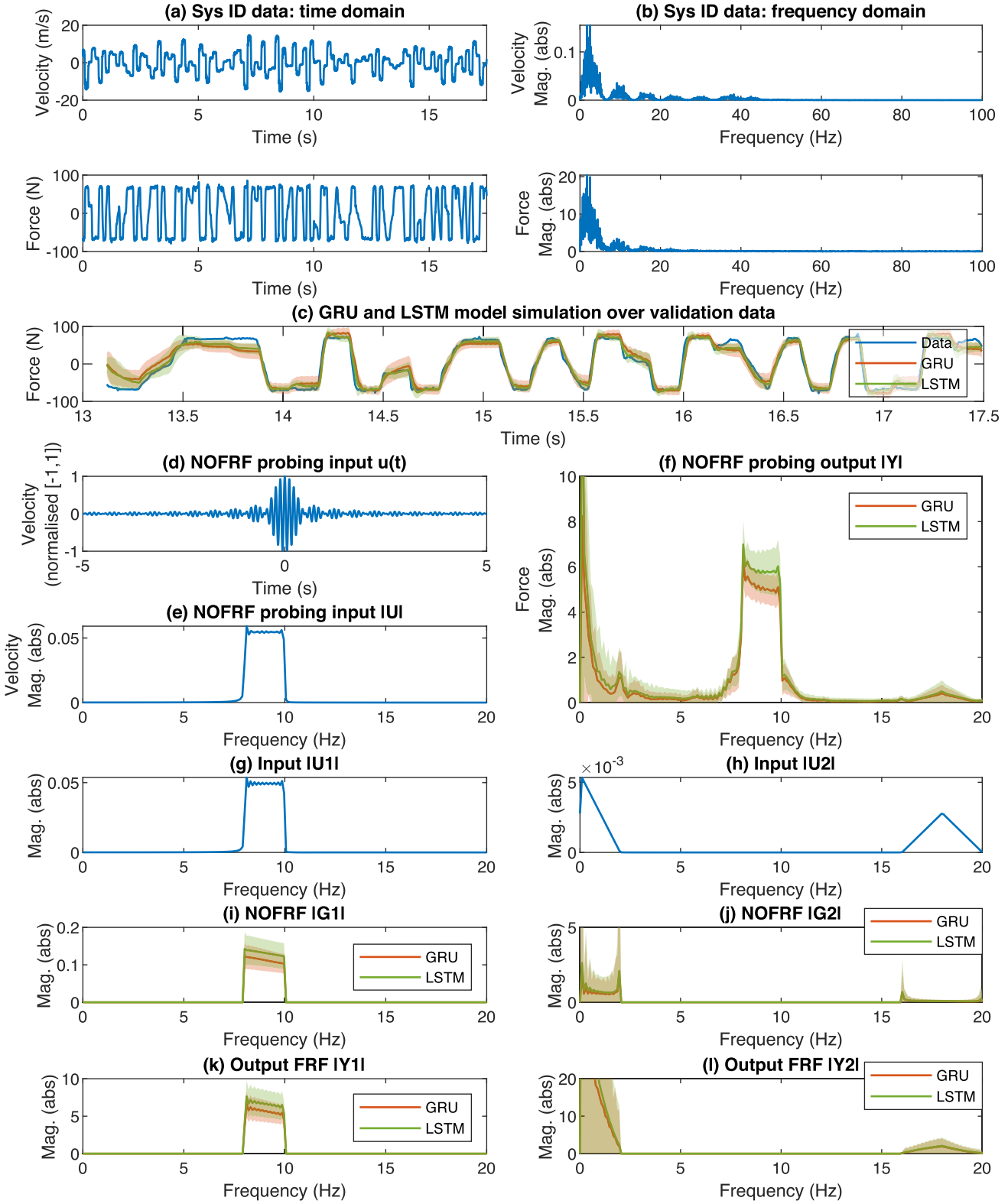

FIGURE 3. MR damper modelling and interpretation using frequency response functions. (a)-(b) System identification input-output data in the time and frequency-domains. (c) GRU and LSTM model predictions compared to validation data. (d)-(e) NOFRF input data used to probe the nonlinear system. (f) NOFRF probing output in the frequency-domain. (g)-(h) First and second order inputs. (i)-(j) First and second order NOFRFs. (k)-(l) First and second order output frequency responses. In each plot, a shaded region indicates the 95% confidence interval derived from uncertainty quantification using bootstrapping.

to interpret model behaviour because they have the advantage of being one-dimensional functions that are therefore simple to analyse graphically. We also enhanced the NOFRF analysis method using uncertainty quantification via the stationary bootstrap method. This was particularly useful for comparing models and revealing the increased uncertainty at low frequencies in the models, compared to high frequencies. The approach was demonstrated on two different systems: a numerical example of a nonlinear mass-spring-damper and a real-world example of a magneto-rheological damper. In both cases the nonlinear systems exhibited effects such as energy transfer between frequencies - the results demonstrated how the NOFRFs could be used to explain why the output exhibited these phenomena, via the input spectra and the filtering effect of the NOFRFs.

To date, NOFRFs have mainly been used to analyse nonlinear systems using NARX models [1], [25] and so the extension to investigating wider classes of deep learning model would be interesting for future work. The approach reported here, for interpreting deep learning models in NSID, is flexible and extensible to other model classes because both the NOFRF analysis and the uncertainty quantification rely on time-domain identification and simulation of the model, therefore the approach can be extended to complex networks with multibranch pathways. In addition, the method for uncertainty quantification used here, the bootstrap, is both simple and flexible, but also computationally intensive. It is worth noting that a study on image classification found that resampling the data was unnecessary and that random sampling of the parameters was sufficient to quantify uncertainty [41], which would be interesting to investigate for recurrent models in NSID problems. Additionally, it would be of interest to investigate how modern variational inference methods in deep learning for NSID [35], [36] could be linked to more efficient methods of uncertainty quantification in the frequency-domain. This has been done for NARX models [45], resulting in significant improvements in computational efficiency, and so highlights a future research gap to address for deep learning models.

V. SUMMARY

In summary, we have demonstrated an approach for deep learning model interpretation in NSID, using frequency response functions combined with uncertainty quantification derived from the bootstrap. The approach was successfully applied to two different nonlinear systems demonstrating that it is possible to gain valuable insight and interpretation of the system in a way that is not possible by just analysing the black box model equations.

VI. DATA AVAILABILITY

The datasets generated during and/or analysed during the current study are available from the corresponding author on reasonable request.

REFERENCES

[1] Z. Q. Lang and S. A. Billings, "Energy transfer properties of non-linear systems in the frequency domain," *Int. J. Control*, vol. 78, no. 5, pp. 345–362, Mar. 2005.

[2] Z. K. Peng, Z. Q. Lang, and S. A. Billings, "Crack detection using nonlinear output frequency response functions," *J. Sound Vib.*, vol. 301, nos. 3–5, pp. 777–788, Apr. 2007.

[3] W. R. Jacobs, E. D. Wilson, T. Assaf, J. Rossiter, T. J. Dodd, J. Porrill, and S. R. Anderson, "Control-focused, nonlinear and time-varying modelling of dielectric elastomer actuators with frequency response analysis," *Smart Mater. Struct.*, vol. 24, no. 5, May 2015, Art. no. 055002.

[4] Y.-P. Zhu, Z. Q. Lang, H.-L. Mao, and H. Laalej, "Nonlinear output frequency response functions: A new evaluation approach and applications to railway and manufacturing systems' condition monitoring," *Mech. Syst. Signal Process.*, vol. 163, Jan. 2022, Art. no. 108179.

[5] S. A. Billings, *Nonlinear System Identification: NARMAX Methods in the Time, Frequency, and Spatio-Temporal Domains*. Hoboken, NJ, USA: Wiley, 2013.

[6] K. Narendra, "Neural networks for control: Theory and practice," *Proc. IEEE*, vol. 84, no. 10, pp. 1385–1406, Oct. 1996.

[7] T. Takagi and M. Sugeno, "Fuzzy identification of systems and its applications to modeling and control," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-15, no. 1, pp. 116–132, Jan. 1985.

[8] J. Sjöberg, Q. Zhang, L. Ljung, A. Benveniste, B. Delyon, P.-Y. Glorennec, H. Hjalmarsson, and A. Juditsky, "Nonlinear black-box modeling in system identification: A unified overview," *Automatica*, vol. 31, no. 12, pp. 1691–1724, Dec. 1995.

[9] T. B. Schön, A. Wills, and B. Ninness, "System identification of nonlinear state-space models," *Automatica*, vol. 47, no. 1, pp. 39–49, Jan. 2011.

[10] T. Baldacchino, S. R. Anderson, and V. Kadirkamanathan, "Structure detection and parameter estimation for NARX models in a unified EM framework," *Automatica*, vol. 48, no. 5, pp. 857–865, May 2012.

[11] L. Ljung, C. Andersson, K. Tiels, and T. B. Schön, "Deep learning and system identification," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 1175–1181, 2020.

[12] W. J. Murdoch, C. Singh, K. Kumbier, R. Abbasi-Asl, and B. Yu, "Definitions, methods, and applications in interpretable machine learning," *Proc. Nat. Acad. Sci. USA*, vol. 116, no. 44, pp. 22071–22080, Oct. 2019.

[13] Z. C. Lipton, "The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery," *Queue*, vol. 16, no. 3, pp. 31–57, Jun. 2018.

[14] C. Molnar, *Interpretable Machine Learning: A Guide for Making Black Box Models Explainable*, 2nd ed. Chicago, IL, USA: Independent, 2022.

[15] D. A. George, "Continuous nonlinear systems," Research Lab. Electron., Massachusetts Inst. Technol., Cambridge, MA, USA, Tech. Rep. 355, 1959.

[16] K. I. Kim and E. J. Powers, "A digital method of modeling quadratically nonlinear systems with a general random input," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 36, no. 11, pp. 1758–1769, Nov. 1988.

[17] C.-H. Tseng and E. J. Powers, "Identification of cubic systems using higher order moments of i.i.d. signals," *IEEE Trans. Signal Process.*, vol. 43, no. 7, pp. 1733–1735, Jul. 1995.

[18] D. E. Adams, "Frequency domain ARX model and multi-harmonic FRF estimators for non-linear dynamic systems," *J. Sound Vib.*, vol. 250, no. 5, pp. 935–950, Mar. 2002.

[19] S. A. Billings and K. M. Tsang, "Spectral analysis for non-linear systems—Part I: Parametric non-linear spectral analysis," *Mech. Syst. Signal Process.*, vol. 3, no. 4, pp. 319–339, Oct. 1989.

[20] J. C. P. Jones and K. Choudhary, "Efficient computation of higher order frequency response functions for nonlinear systems with, and without, a constant term," *Int. J. Control*, vol. 85, no. 5, pp. 578–593, May 2012.

[21] P. Palumbo and L. Piroddi, "Harmonic analysis of non-linear structures by means of generalised frequency response functions coupled with NARX models," *Mech. Syst. Signal Process.*, vol. 14, no. 2, pp. 243–265, Mar. 2000.

[22] E. Bedrosian and S. O. Rice, "The output properties of Volterra systems (nonlinear systems with memory) driven by harmonic and Gaussian inputs," *Proc. IEEE*, vol. 59, no. 12, pp. 1688–1707, Dec. 1971.

[23] J. C. P. Jones and S. A. Billings, "Recursive algorithm for computing the frequency response of a class of non-linear difference equation models," *Int. J. Control*, vol. 50, no. 5, pp. 1925–1940, Nov. 1989.

[24] M. Schetzen, *The Volterra and Wiener Theories of Nonlinear Systems*. New York, NY, USA: Wiley, 1980.

[25] R. S. Bayma, Y. Zhu, and Z.-Q. Lang, "The analysis of nonlinear systems in the frequency domain using nonlinear output frequency response functions," *Automatica*, vol. 94, pp. 452–457, Aug. 2018.
[26] L. Chen, Z. Zhang, J. Cao, and X. Wang, "A novel method of combining nonlinear frequency spectrum and deep learning for complex system fault diagnosis," *Measurement*, vol. 151, Feb. 2020, Art. no. 107190.
[27] L. Chen, H. Hu, Z. Zhang, and X. Wang, "Application of nonlinear output frequency response functions and deep learning to RV reducer fault diagnosis," *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1–14, 2021.
[28] B. Zhao, C. Cheng, Z. Peng, X. Dong, and G. Meng, "Detecting the early damages in structures with nonlinear output frequency response functions and the CNN-LSTM model," *IEEE Trans. Instrum. Meas.*, vol. 69, no. 12, pp. 9557–9567, Dec. 2020.
[29] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder–decoder for statistical machine translation," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2014, pp. 1724–1734.
[30] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.
[31] W. R. Jacobs, T. Baldacchino, T. Dodd, and S. R. Anderson, "Sparse Bayesian nonlinear system identification using variational inference," *IEEE Trans. Autom. Control*, vol. 63, no. 12, pp. 4172–4187, Dec. 2018.
[32] T. Baldacchino, S. R. Anderson, and V. Kadirkamanathan, "Computational system identification for Bayesian NARMAX modelling," *Automatica*, vol. 49, no. 9, pp. 2641–2651, Sep. 2013.
[33] P. L. Green and K. Worden, "Bayesian and Markov chain Monte Carlo methods for identifying nonlinear systems in the presence of uncertainty," *Phil. Trans. Roy. Soc. A, Math., Phys. Eng. Sci.*, vol. 373, no. 2051, Sep. 2015, Art. no. 20140405.
[34] M. Abdar, F. Pourpanah, S. Hussain, D. Rezazadegan, L. Liu, M. Ghavamzadeh, P. Fieguth, X. Cao, A. Khosravi, U. R. Acharya, V. Makarenkov, and S. Nahavandi, "A review of uncertainty quantification in deep learning: Techniques, applications and challenges," *Inf. Fusion*, vol. 76, pp. 243–297, Dec. 2021.
[35] A. Brusaferri, M. Matteucci, P. Portolani, and S. Spinelli, "Nonlinear system identification using a recurrent network in a Bayesian framework," in *Proc. IEEE 17th Int. Conf. Ind. Informat. (INDIN)*, Jul. 2019, pp. 319–324.
[36] H. Zhou, C. Ibrahim, W. X. Zheng, and W. Pan, "Sparse Bayesian deep learning for dynamic system identification," *Automatica*, vol. 144, Oct. 2022, Art. no. 110489.
[37] T. Chen, E. Fox, and C. Guestrin, "Stochastic gradient Hamiltonian Monte Carlo," in *Proc. Int. Conf. Mach. Learn.*, 2014, pp. 1683–1691.
[38] W. J. Maddox, P. Izmailov, T. Garipov, D. P. Vetrov, and A. G. Wilson, "A simple baseline for Bayesian uncertainty in deep learning," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, 2019.
[39] Y. Gal and Z. Ghahramani, "A theoretically grounded application of dropout in recurrent neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 29, 2016.
[40] S. Semeniuta, A. Severyn, and E. Barth, "Recurrent dropout without memory loss," in *Proc. 26th Int. Conf. Comput. Linguistics, Tech. Papers*, 2016, pp. 1757–1766.
[41] B. Lakshminarayanan, A. Pritzel, and C. Blundell, "Simple and scalable predictive uncertainty estimation using deep ensembles," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017.
[42] Y. Wen, D. Tran, and J. Ba, "BatchEnsemble: An alternative approach to efficient ensemble and lifelong learning," in *Proc. Int. Conf. Learn. Represent.*, 2020. [Online]. Available: https://openreview.net/forum?id=Sklf1yrYDr
[43] W. G. Martinez, "Ensemble pruning via quadratic margin maximization," *IEEE Access*, vol. 9, pp. 48931–48951, 2021.
[44] K. Worden, "Confidence bounds for frequency response functions from time series models," *Mech. Syst. Signal Process.*, vol. 12, no. 4, pp. 559–569, Jul. 1998.
[45] W. R. Jacobs, T. J. Dodd, and S. R. Anderson, "Frequency-domain analysis for nonlinear systems with time-domain model parameter uncertainty," *IEEE Trans. Autom. Control*, vol. 64, no. 5, pp. 1905–1915, May 2019.
[46] B. Efron, "Bootstrap methods: Another look at the jackknife," *Ann. Statist.*, vol. 7, no. 1, pp. 1–26, Jan. 1979.
[47] B. Efron and R. J. Tibshirani, *An Introduction to the Bootstrap*. Boca Raton, FL, USA: CRC Press, 1994.
[48] J. Berkowitz and L. Kilian, "Recent developments in bootstrapping time series," *Econ. Res.*, vol. 19, no. 1, pp. 1–48, 2000.
[49] A. M. Zoubir and D. R. Iskandler, "Bootstrap methods and applications," *IEEE Signal Process. Mag.*, vol. 24, no. 4, pp. 10–19, Jul. 2007.
[50] H. L. Wei and S. A. Billings, "Improved parameter estimates for non-linear dynamical models using a bootstrap method," *Int. J. Control*, vol. 82, no. 11, pp. 1991–2001, Nov. 2009.
[51] S. Shanta and V. Kadirkamanathan, "A bootstrap-based approach for parameter and polyspectral density estimation of a non-minimum phase ARMA process," *Int. J. Syst. Sci.*, vol. 46, no. 3, pp. 418–428, Feb. 2015.
[52] D. N. Politis and J. P. Romano, "The stationary bootstrap," *J. Amer. Statist. Assoc.*, vol. 89, no. 428, pp. 1303–1313, 1994.
[53] I. J. Leontaritis and S. A. Billings, "Experimental design and identifiability for non-linear systems," *Int. J. Syst. Sci.*, vol. 18, no. 1, pp. 189–202, Jan. 1987.
[54] L. Ljung, *System Identification: Theory for the User*, 2nd ed. Upper Saddle River, NJ, USA: Prentice-Hall, 1999.
[55] O. Nelles and R. Isermann, "Identification of nonlinear dynamic systems classical methods versus radial basis function networks," in *Proc. Amer. Control Conf. (ACC)*, vol. 5, Jun. 1995, pp. 3786–3790.
[56] S. A. Billings, S. Chen, and M. J. Korenberg, "Identification of MIMO non-linear systems using a forward-regression orthogonal estimator," *Int. J. Control*, vol. 49, no. 6, pp. 2157–2189, Jun. 1989.
[57] L. Piroddi and W. Spinelli, "An identification algorithm for polynomial NARX models based on simulation error minimization," *Int. J. Control*, vol. 76, no. 17, pp. 1767–1781, Nov. 2003.
[58] K. Z. Mao and S. A. Billings, "Algorithms for minimal model structure detection in nonlinear dynamic system identification," *Int. J. Control*, vol. 68, no. 2, pp. 311–330, Jan. 1997.
[59] J. Platt, "A resource-allocating network for function interpolation," *Neural Comput.*, vol. 3, no. 2, pp. 213–225, Jun. 1991.
[60] V. Kadirkamanathan and M. Niranjan, "A function estimation approach to sequential learning with neural networks," *Neural Comput.*, vol. 5, no. 6, pp. 954–975, Nov. 1993.
[61] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.
[62] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization," *J. Mach. Learn. Res.*, vol. 13, no. 2, pp. 281–305, Feb. 2012.
[63] K. Greff, R. K. Srivastava, J. Koutník, B. R. Steunebrink, and J. Schmidhuber, "LSTM: A search space Odyssey," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 10, pp. 2222–2232, Oct. 2017.
[64] J. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl, "Algorithms for hyper-parameter optimization," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 24, 2011.
[65] J. Snoek, H. Larochelle, and R. P. Adams, "Practical Bayesian optimization of machine learning algorithms," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 25, 2012.
[66] P. J. Angeline, G. M. Saunders, and J. B. Pollack, "An evolutionary algorithm that constructs recurrent neural networks," *IEEE Trans. Neural Netw.*, vol. 5, no. 1, pp. 54–65, Jan. 1994.
[67] T. Elsken, J. H. Metzen, and F. Hutter, "Neural architecture search: A survey," *J. Mach. Learn. Res.*, vol. 20, no. 1, pp. 1997–2017, 2019.
[68] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. 3rd Int. Conf. Learn. Represent. (ICLR)*, Y. Bengio and Y. LeCun, Eds., San Diego, CA, USA, May 2015.
[69] P. J. Werbos, "Backpropagation through time: What it does and how to do it," *Proc. IEEE*, vol. 78, no. 10, pp. 1550–1560, Oct. 1990.
[70] O. Nelles, *Nonlinear System Identification*. Berlin, Germany: Springer, 2001.
[71] H. R. Kunsch, "The jackknife and the bootstrap for general stationary observations," *Ann. Statist.*, vol. 17, no. 3, pp. 1217–1241, Sep. 1989.
[72] Y. Liu, Y. L. Zhao, J. T. Li, H. Ma, Q. Yang, and X. X. Yan, "Application of weighted contribution rate of nonlinear output frequency response functions to rotor rub-impact," *Mech. Syst. Signal Process.*, vol. 136, Feb. 2020, Art. no. 106518.
[73] K. Worden, P. K. Stansby, G. R. Tomlinson, and S. A. Billings, "Identification of nonlinear wave forces," *J. Fluids Struct.*, vol. 8, no. 1, pp. 19–71, Jan. 1994.
[74] J. Wang, A. Sano, T. Chen, and B. Huang, "Identification of Hammerstein systems without explicit parameterisation of non-linearity," *Int. J. Control*, vol. 82, no. 5, pp. 937–952, May 2009.
[75] L. Ljung and R. Singh, "Version 8 of the MATLAB system identification toolbox," *IFAC Proc. Volumes*, vol. 45, no. 16, pp. 1826–1831, Jul. 2012.

WILL R. JACOBS received the M.Phys. degree in physics with mathematics and the Ph.D. degree in nonlinear system identification from The University of Sheffield, Sheffield, U.K., in 2010 and 2016, respectively.

He is currently a Research Associate with the Department of Automatic Control and Systems Engineering, Rolls-Royce University Technology Centre, The University of Sheffield.

VISAKAN KADIRKAMANATHAN (Member, IEEE) received the B.A. degree in electrical and information sciences and the Ph.D. degree in information engineering from the Department of Engineering, University of Cambridge, Cambridge, U.K., in 1987 and 1992, respectively.

Following brief postdoctoral research positions with the University of Surrey, Guildford, U.K., and the University of Cambridge, he was appointed as a Lecturer with the Department of Automatic Control and Systems Engineering, The University of Sheffield, Sheffield, U.K. He is currently a Professor of signal and information processing and the Director of the Rolls-Royce University Technology Centre for Control, Monitoring, and Systems Engineering, in the Department of Automatic Control and Systems Engineering, Sheffield. He has published more than 200 papers in peer-reviewed journals and conferences. His research interests are in the areas of data-driven modeling, signal processing, and control, with applications in aerospace, biomedical, and other dynamic systems.

Dr. Kadirkamanathan was a recipient of the 2012 Proceedings of the National Academy of Science (PNAS) Cozzarelli Prize, USA. He is a past Associate Editor of IEEE TRANSACTIONS ON NEURAL NETWORKS, among other conference and journal activities.

SEAN R. ANDERSON received the M.Eng. degree in control systems engineering and the Ph.D. degree in nonlinear system identification from The University of Sheffield, Sheffield, U.K., in 2001 and 2005, respectively.

Then, he was a Postdoctoral Researcher in computational neuroscience, bioinspired robotics, and signal processing with The University of Sheffield, from 2005 to 2011. He joined the Department of Automatic Control and Systems Engineering, The University of Sheffield, as a Lecturer, in 2012, and has been a Senior Lecturer, since 2015, working in the areas of robotics, nonlinear system identification, and computational biology.