ETH zürich

Feature-centric nonlinear autoregressive models

S. Schär, S. Marelli and B. Sudret

Chair of Risk, Safety and Uncertainty Quantification
Stefano-Franscini-Platz 5
CH-8093 Zürich

Risk, Safety & Uncertainty Quantification

Data Sheet

Journal:	Submitted
Report Ref.:	RSUQ-2024-009A
Arxiv Ref.:	https://arxiv.org/abs/2410.07293 [stat.ME] [stat.AP]
DOI:	-
Date submitted:	September 26, 2024
Date accepted:	-

Feature-centric nonlinear autoregressive models

Styfen Schär [*1], Stefano Marelli[†1], and Bruno Sudret[‡1]

[1] *Chair of Risk, Safety and Uncertainty Quantification, ETH Zürich, Switzerland*

October 9, 2024

Abstract

We propose a novel feature-centric approach to surrogate modeling of dynamical systems driven by time-varying exogenous excitations. This approach, named Functional Nonlinear AutoRegressive with eXogenous inputs ($\mathcal{F}$-NARX), aims to approximate the system response based on temporal features of both the exogenous inputs and the system response, rather than on their values at specific time lags. This is a major step away from the discrete-time-centric approach of classical NARX models, which attempts to determine the relationship between selected time steps of the input/output time series. By modeling the system in a time-feature space instead of the original time axis, $\mathcal{F}$-NARX can provide more stable long-term predictions and drastically reduce the reliance of the model performance on the time discretization of the problem.

$\mathcal{F}$-NARX, like NARX, acts as a framework and is not tied to a single implementation. In this work, we introduce an $\mathcal{F}$-NARX implementation based on principal component analysis and polynomial basis functions. To further improve prediction accuracy and computational efficiency, we also introduce a strategy to identify and fit a sparse model structure, thanks to a modified hybrid least angle regression approach that minimizes the expected forecast error, rather than the one-step-ahead prediction error.

Since $\mathcal{F}$-NARX is particularly well-suited to modeling engineering structures typically governed by physical processes, we investigate the behavior and capabilities of our $\mathcal{F}$-NARX implementation on two case studies: an eight-story building under wind loading and a three-story steel frame under seismic loading. While the first case study highlights the simple and intuitive parametrization of the presented $\mathcal{F}$-NARX implementation, the second case study demonstrates its high accuracy and prediction stability in more complex nonlinear problems.

Our results demonstrate that $\mathcal{F}$-NARX has several favorable properties over classical NARX, making it well suited to emulate engineering systems with high accuracy over extended time periods.

[*] styfen.schaer@ibk.baug.ethz.ch

[†] marelli@ibk.baug.ethz.ch

[‡] sudret@ethz.ch

1 Introduction

Many real-world problems are inherently time-dependent, with systems continuously evolving under the influence of external factors and excitations. Accurate modeling of these dynamical systems is essential in numerous engineering disciplines, be it for system control Levin and Narendra (1996); Hu et al. (2024), maintenance planning Langeron et al. (2021); Samsuri et al. (2023), fault detection and diagnosis Mattson and Pandit (2006); Gao et al. (2016), design assessment and optimization Yu et al. (2023); Deshmukh and Allison (2017) or uncertainty quantification Mai et al. (2016); Bhattacharyya et al. (2020) and reliability analysis Garg et al. (2022); Zhou and Li (2023); Zhang et al. (2024).

Although these applications are different in nature, they are similar in that they often require to learn the dependence between the time-varying external factors or actions and the corresponding system response. This dependence is typically modeled by so-called autoregressive models with exogenous inputs (ARX). By using the exogenous inputs in conjunction with their own past predictions, these models are powerful predictors for dynamical systems, especially when used in their nonlinear variant, known as NARX Billings (2013). NARX models have been successfully applied to many engineering structures and components, such as mooring lines Yetkin et al. (2017); Zhang et al. (2024), gas or wind turbines Chiras et al. (2001); Schlechtingen and Ferreira Santos (2011), multi-story buildings Spiridonakos and Chatzi (2015); Li et al. (2021), marine structures Kim (2015); Ramin et al. (2023) or geotechnical structures Wunsch et al. (2018); Dassanayake et al. (2023).

This success is partially due to their versatility, which allows one to combine them with powerful techniques from the fields of surrogate modeling and machine learning, such as Gaussian process modeling Murray-Smith et al. (1999); Kocijan (2012); Koziel et al. (2014); Worden et al. (2018), support vector regression Acuña et al. (2012); Ranković et al. (2014); Zhang et al. (2017), neural networks Siegelmann et al. (1997); Li et al. (2021); Song et al. (2022) or a wide array of basis functions Aguirre and Billings (1993); Coca and Billings (2001); Chen et al. (2008). They have also been successfully deployed in applications with categorical outcomes by combining them with logistic regression Ayala et al. (2017); Lacerda Junior et al. (2021).

Fitting a predictive model to existing data is generally a difficult task, and since NARX models rely on the time discretization of the problem at hand, they pose one additional key challenge: the selection of lags to be considered. In other words, how to select which time instants of the exogenous and autoregressive inputs are to be considered in the model? This selection has to be made from a possibly large set of candidate lags, for example, if the time-discretization of the problem is small or the model's memory long. While no universal solution to this problem is currently available from the relevant literature, several ways to directly select important lags, or at least decimate the candidate lags, have been proposed. Examples include trial-and-error-based approaches Chen and Ni (2011); Schär et al. (2024); Awtoniuk et al. (2019) or correlation-based approaches Wei and Billings (2008); Cheng et al. (2011). An approach named "clustering" has

been proposed by Aguirre and Billings (1995); Aguirre and Jácome (1998) for the specific case of polynomial NARX models. For the special class of *linear-in-the-parameters* problems Billings (2013), dedicated sparse solvers have also been employed to better handle a large number of regressors Spinelli et al. (2006); Billings (2013); Falsone et al. (2014); Yuzhu Guo and Wei (2015); Mai et al. (2016); F. Bianchi and Piroddi (2017); Li et al. (2021).

NARX models are most often trained by minimizing the one-step-ahead prediction (OSA) error, as this is very fast and sometimes even a closed-form solution exists. However, minimization of OSA errors can lead to poor results when the trained model is used in a forecast setup Piroddi (2008). In extreme cases, the model predictions can even diverge over time Piroddi (2008); Farina and Piroddi (2009); Yu et al. (2023). Therefore, dedicated solvers have been developed to minimize the forecast error. For instance, Piroddi and Spinelli (2003); Piroddi (2008); Piroddi et al. (2010) present pruning-based methods, while Farina and Piroddi (2009, 2010) introduced gradient-based algorithms. The main disadvantage of these approaches is the heavily increased computational costs compared to the minimization of the OSA error.

Several works Aguirre (1994); Billings and Aguirre (1995); Spinelli et al. (2006) argue that the negative consequences of an OSA error minimization are more likely to happen if the sampling frequency of the problem is high. The reason lies in the high temporal correlation between adjacent time steps, which in turn can lead to the over-reliance on the first autoregressive lag, resulting in drastic losses in terms of the final model performance Piroddi and Spinelli (2003).

Interestingly, all these limitations are related to the discretization of the time axis, in conjunction with the exclusive focus of classical ARX modeling on individual discrete time step values, a characteristic we refer to as a *discrete-time* view on ARX modeling. Although many methods have been developed to mitigate these problems, they remain fully committed to this discrete-time view. This raises the question of whether alternative views can offer advantages over the discrete-time one.

In this work, we revisit the fundamentals of autoregressive problems with exogenous inputs. We propose and discuss an alternative *continuous functional* view, demonstrating how it can address the problem of lag selection, while at the same time mitigating the problem of over-reliance, thus improving forecast stability and accuracy. This continuous functional approach leverages the temporal correlation and smoothness exhibited by many time-dependent processes, especially in physical systems or structural responses. Due to its underlying principles, we refer to this approach as *functional nonlinear autoregressive with exogenous inputs* ($\mathcal{F}$-NARX) modeling. The way $\mathcal{F}$-NARX modeling uses temporal features of the input and output signals allows the problem to still be posed as a linear regression problem, and therefore $\mathcal{F}$-NARX remains compatible with most algorithms developed for classical NARX modeling. The use of temporal features can also result in sparser representations of both the input and output signals, and promotes independence between regressors, which can in turn be exploited through compressed sensing algorithms Eldar and Kutyniok (2012), and thus create small models with high expressiveness.

This paper is structured as follows: we begin with the description of the $\mathcal{F}$-NARX approach in the Sections 2 and 3. This includes a brief recap on the basics of NARX modeling, followed by its extension to $\mathcal{F}$-NARX, as well as a concrete implementation of an $\mathcal{F}$-NARX model using a combination of principal-component analysis and a polynomial regression model. In Section 4, we present two case studies to investigate the behavior and properties of the presented $\mathcal{F}$-NARX implementation and demonstrate its performance on a complex dynamical system. A discussion on the $\mathcal{F}$-NARX approach and the presented results, as well as concluding remarks, are given in Section 5.

2 Methodology

2.1 Autoregressive modeling with exogenous inputs

Consider a deterministic dynamical system $\mathcal{M}$ evolving along the time axis $\mathcal{T}$. The system is excited by a (possibly high-dimensional) time-varying exogenous input $\boldsymbol{x}(t) \in \mathbb{R}^M$. Given a vector of initial conditions $\boldsymbol{\beta}$, the system response $y(t) \in \mathbb{R}$ at any time instant $t \in \mathcal{T}$ is denoted as:

$$y(t) = \mathcal{M}(\boldsymbol{x}(\mathcal{T} \leq t), \boldsymbol{\beta}). \tag{1}$$

Here, the notation $\bullet(\mathcal{T} \leq t)$ indicates that the system response at time t depends on the excitation up to and including time t. To simplify the notation, we will subsequently omit $\boldsymbol{\beta}$ unless it is strictly necessary.

Suppose we want to construct a dynamic surrogate model $\widehat{\mathcal{M}}$ that approximates the response of the system $\mathcal{M}$ such that:

$$\widehat{y}(t) = \widehat{\mathcal{M}}(\boldsymbol{x}(\mathcal{T} \leq t), \boldsymbol{\beta}) \approx \mathcal{M}(\boldsymbol{x}(\mathcal{T} \leq t), \boldsymbol{\beta}). \tag{2}$$

In this work, we focus in particular on dynamic surrogates based on nonlinear autoregressive models with exogenous inputs (NARX). At the core of a NARX model lies the idea that the system's dynamics can be captured at a set of discrete time steps $\{0, \delta t, \ldots, (N-1)\delta t\}$ and that the system response at a time step in the near future can be predicted as a function of past and current exogenous inputs, as well as past outputs:

$$\widehat{y}(t + \delta t) = \widehat{\mathcal{M}}(\boldsymbol{x}(\mathcal{T} \leq t + \delta t), y(\mathcal{T} < t + \delta t); \boldsymbol{c}) + \varepsilon(t), \tag{3}$$

where $\varepsilon(t) \sim \mathcal{N}(0, \sigma_\varepsilon(t))$ is a residual term with zero mean and standard deviation $\sigma_\varepsilon(t)$. Let us assume the mapping function $\widehat{\mathcal{M}}$ is parametric, and that it can be characterized by a finite set of parameters $\boldsymbol{c}$. Then the process of training (or fitting) a NARX model consists in estimating the values of $\boldsymbol{c}$ from a set of system input/output discretized trajectories $\left(\boldsymbol{x}^{(i)}, \boldsymbol{y}^{(i)}\right)$. We refer to each such pair as a *realization* or *observation* of the system, while the full set of realizations is called the *experimental design* (ED):

$$\mathcal{D} = \left\{ \left(\boldsymbol{x}^{(i)}, \boldsymbol{y}^{(i)}\right), \boldsymbol{x}^{(i)} \in \mathbb{R}^{N \times M}, \boldsymbol{y}^{(i)} = \mathcal{M}(\boldsymbol{x}^{(i)}) \in \mathbb{R}^N, i = 1, \ldots, N_{\text{ED}} \right\}. \tag{4}$$

The number of realizations in the experimental design is usually relatively small ($\mathcal{O}(10^{1-2})$), since the acquisition of even a single observation may require an expensive experiment or simulation.

We can make the notation more explicit by rewriting Eq. (3) as:

$$\widehat{y}(t+\delta t) = \widehat{\mathcal{M}}(\boldsymbol{\varphi}(t+\delta t); \boldsymbol{c}), \tag{5}$$

where the vector $\boldsymbol{\varphi}(t) \in \mathbb{R}^n$ reads:

$$\begin{aligned} \boldsymbol{\varphi}(t) = \{ & y(t-\delta t), y(t-2\delta t), \dots, y(t-n_y\delta t), \\ & x_1(t), x_1(t-\delta t), \dots, x_1(t-n_{x_1}\delta t), \\ & \dots, \\ & x_M(t), x_M(t-\delta t), \dots, x_M(t-n_{x_M}\delta t)\}. \end{aligned} \tag{6}$$

We refer to the delayed values $y(t-(k+1)\delta t)$ as the autoregressive lags and to $x_i(t-k\delta t)$ as the exogenous input lags, where k is a non-negative integer value to preserve the causality of the original system. The integers $\{n_y, n_{x_1}, \dots, n_{x_M}\}$ are typically referred to as the *model orders*.

By stacking the vectors $\boldsymbol{\varphi}(t)$ for all time steps, we obtain the so-called *design matrix* $\boldsymbol{\Phi} \in \mathbb{R}^{\widetilde{N}\times n}$, where $n = n_y + n_{x_1} + \cdots + n_{x_M}$ and where $\widetilde{N} \leq N$ due to the use of lagged values. Similarly, we stack the values $y(t)$ to obtain the corresponding output vector $\boldsymbol{y} \in \mathbb{R}^{\widetilde{N}}$:

$$\boldsymbol{\Phi} = \begin{pmatrix} \boldsymbol{\varphi}(t_0) \\ \boldsymbol{\varphi}(t_0+\delta t) \\ \vdots \\ \boldsymbol{\varphi}(t_0+(N-1)\delta t) \end{pmatrix}, \quad \boldsymbol{y} = \begin{pmatrix} y(t_0) \\ y(t_0+\delta t) \\ \vdots \\ y(t_0+(N-1)\delta t) \end{pmatrix}, \tag{7}$$

with $t_0 = \max(n_y, n_{x_1}, \dots, n_{x_M})\delta t$.

Note that despite considering time-dependent data, the sample pairs $\{\boldsymbol{\varphi}(t), y(t)\}$ no longer need to follow a temporal ordering. Consequently, the matrices $\boldsymbol{\Phi}^{(i)}$ and vectors $\boldsymbol{y}^{(i)}$, with $i = 1, \dots, N_{\text{ED}}$, from different realizations within the experimental design can be concatenated to form a larger matrix $\boldsymbol{\Phi}_{\text{ED}}$ and vector $\boldsymbol{y}_{\text{ED}}$:

$$\boldsymbol{\Phi}_{\text{ED}} = \begin{pmatrix} \boldsymbol{\Phi}^{(1)} \\ \vdots \\ \boldsymbol{\Phi}^{(N_{\text{ED}})} \end{pmatrix}, \quad \boldsymbol{y}_{\text{ED}} = \begin{pmatrix} \boldsymbol{y}^{(1)} \\ \vdots \\ \boldsymbol{y}^{(N_{\text{ED}})} \end{pmatrix}. \tag{8}$$

We now recall from Eq. (3), that the objective is to develop a predictive model $\widehat{\mathcal{M}}$, characterized by its parameters $\boldsymbol{c}$, using only a limited set of observations $\mathcal{D}$. The estimation of these model parameters can then be performed by minimizing a suitable loss function $\mathcal{L}$ that represents the discrepancy between the experimental design observations and the model predictions:

$$\hat{\boldsymbol{c}} = \arg\min_{\boldsymbol{c}} \mathcal{L}\left(\boldsymbol{y}_{\text{ED}}, \widehat{\mathcal{M}}(\boldsymbol{\Phi}_{\text{ED}}; \boldsymbol{c})\right). \tag{9}$$

Common choices for $\widehat{\mathcal{M}}$ include polynomials, for which $\boldsymbol{c}$ represents the coefficient vector Aguirre and Billings (1993), neural networks, with $\boldsymbol{c}$ denoting the corresponding weights and biases Siegelmann et al. (1997), among others. A class of models frequently adopted in the literature is that of linear-in-the-parameters models Billings (2013). This class is particularly attractive, because it allows the optimization problem in Eq. (9) to be solved using least-squares minimization:

$$\hat{\boldsymbol{c}} = \arg\min_{\boldsymbol{c}} \|\boldsymbol{y}_{\mathrm{ED}} - \mathcal{G}(\boldsymbol{\Phi}_{\mathrm{ED}})\boldsymbol{c}\|^2, \tag{10}$$

where $\mathcal{G} : \mathbb{R}^{|\boldsymbol{\Phi}|} \to \mathbb{R}^{|\boldsymbol{c}|}$ denotes a possibly nonlinear mapping between lags and regressors, e. g. multivariate polynomials. By defining $\boldsymbol{\Psi}_{\mathrm{ED}} = \mathcal{G}(\boldsymbol{\Phi}_{\mathrm{ED}})$ the coefficient vector $\boldsymbol{c}$ can therefore be computed analytically using ordinary least squares (OLS):

$$\boldsymbol{c} = \left(\boldsymbol{\Psi}_{\mathrm{ED}}{}^{\top}\boldsymbol{\Psi}_{\mathrm{ED}}\right)^{-1}\boldsymbol{\Psi}_{\mathrm{ED}}{}^{\top}\boldsymbol{y}_{\mathrm{ED}}. \tag{11}$$

Although OLS is a viable option to solve for $\boldsymbol{c}$, more advanced regularized regression techniques are often used to promote a sparse set of coefficients, resulting in a more stable model with improved generalization capabilities. A prominent example is given by LASSO (least absolute shrinkage and selection operator) regression Tibshirani (1996) which promotes sparsity introducing an ℓ^1 penalty term with regularization parameter γ for the estimation of the coefficient vector:

$$\hat{\boldsymbol{c}} = \arg\min_{\boldsymbol{c}} \|\boldsymbol{y}_{\mathrm{ED}} - \boldsymbol{\Psi}_{\mathrm{ED}}\boldsymbol{c}\|_2^2 + \gamma||\boldsymbol{c}||_1. \tag{12}$$

Among the available solvers of the LASSO problem, least angle regression (LARS, Efron et al. (2004)) has been proven particularly effective in the context of polynomial regression (Blatman and Sudret, 2011), and is the technique of choice in this work (see Section 3.3).

A detailed explanation and discussion on polynomial NARX models, as a concrete implementation of a linear-in-the-parameters model, and their synergy with sparse regression techniques will be provided in Sections 3.2 and 3.3.

Considering again the linear-in-the-parameters model with its mapping $\mathcal{G}$ and coefficient set $\boldsymbol{c}$, we can calculate the so-called *one-step-ahead* (OSA) prediction on a new input vector $\boldsymbol{\varphi}(t + \delta t)$ as:

$$\widehat{y}(t + \delta t) = \mathcal{G}(\boldsymbol{\varphi}(t + \delta t))\boldsymbol{c}. \tag{13}$$

Note, however, that in order to construct $\boldsymbol{\varphi}(t+\delta t)$, we require the true model output up to time t (see Eq. (6)). While many applications of autoregressive modeling are focused on one-step-ahead predictions, in surrogate modeling this is typically not the case. With a surrogate model, we aim to approximate the true model response *throughout the duration* of the external dynamic loading, i. e. the whole trajectory. We refer to this as *model forecast*, or simply as *model prediction*. The model forecast is based on iteratively evaluating one-step-ahead predictions, and using the corresponding approximate response $\hat{y}(t)$ as the autoregressive input for the next step, resulting

in:

$$\begin{aligned}\widehat{\boldsymbol{\varphi}}(t+\delta t) = \{&\widehat{y}(t), \widehat{y}(t-\delta t), \ldots, \widehat{y}(t-(n_y-1)\delta t),\\ &x_1(t+\delta t), x_1(t), \ldots, x_1(t-(n_{x_1}-1)\delta t),\\ &\ldots,\\ &x_M(t+\delta t), x_M(t), \ldots, x_M(t-(n_{x_M}-1)\delta t)\},\end{aligned} \tag{14}$$

which contains the previous predictions $\widehat{y}(\mathcal{T} \leq t)$. In order to perform this prediction, the model needs to be initialized with values that match the initial conditions of the system under investigation. This is done by setting the n_y initial values of the system response to appropriate values. The surrogate prediction can be initialized to zero Worden and Barthorpe (2012) or other sensible values Schär et al. (2024), depending on the specific application.

2.2 Limitations of classical ARX modeling

If a nonlinear ARX (NARX) model is fitted as described in Eq. (9) or (10), the associated regression problem becomes intractable if the number of exogenous inputs M or the model orders are large. This is due to the highly non-linear scaling of the number of regressors with the problem dimension, a phenomenon known as the *curse of dimensionality* Verleysen and François (2005). To handle systems requiring large model orders, while still keeping the dimensionality of $\boldsymbol{\Phi}$ manageable, Schär et al. (2024) introduced the use of a non-evenly spaced set of lags, determined through trial and error on the available training data. Although it was shown to be a viable option in terms of the accuracy of the final model, trial and error can be time consuming and does not provide a good guarantee of near-optimal performance. Alternatively, Awtoniuk et al. (2019) proposed to select a subset of the lags following certain pre-determined patterns, for example only odd or even lags. However, this approach is limited because regularly spaced lags can cause loss of important information or aliasing effects if significant decimation is performed.

An approach limited to linear-in-the-parameters models, as described in Eq. (10), involves the use of sparse regression solvers. In particular, several forward selection algorithms have gained popularity Billings (2013); Yuzhu Guo and Wei (2015); Mai et al. (2016). Unfortunately, these sparse solvers also have practical limitations regarding the number of regressors. With a very large number of regressors, efficiency decreases, and computational costs or hardware resources can become a bottleneck.

A more specific method, called *term clustering*, was developed for polynomial NARX models to discard entire clusters of lagged values belonging to irrelevant nonlinearities Aguirre and Billings (1995); Aguirre and Jácome (1998); Pulecchi and Piroddi (2007). Selecting the correct clusters can also be particularly challenging, especially in the presence of noise. The specificity to a certain class of NARX models may also have hindered the widespread adoption of this method.

Another difficulty arises when fitting ARX models in a regression setting when the time increment δt is small, e. g. due to requirements of the numerical solver of choice. Very small δt can lead to

numerical instability during the regression task, due to high correlation between lags. In Piroddi and Spinelli (2003) it is shown how a very high sampling rate can also lead to an overestimation of the importance of the most recent autoregressive lag. This *over-reliance* can have a detrimental effect on the model forecast. Unfortunately, data decimation to alleviate these issues is not always viable, since the data may be correctly sampled for signal reconstruction purposes and may only appear oversampled with respect to the prediction problem Piroddi and Spinelli (2003); Lataniotis et al. (2020).

We observe that a common cause of all these limitations lies in the explicitly *discrete-time-centric* nature of classical autoregressive modeling, which assumes a specific time axis discretization, rather than the intrinsically continuous nature of the response of dynamical systems to exogenous excitations. In the following sections, we therefore propose and discuss a novel *continuous-functional* view of the ARX methodology, which tackles autoregressive problems from a continuous perspective and discretizes the problem only in a final stage for numerical purposes.

2.3 Moving from a discrete time to a continuous functional view of autoregressive modeling

Real-world processes are inherently continuous in time, and the discretization of the time axis is merely a tool required by digital data storage or numerical simulators. This fact is acknowledged, for example, by continuous-time system identification, which treats the dynamics of the system as a differential equation Billings (2013). Interestingly, NARX models also find their application in this area. They are used as intermediate discrete parametric models, which are then used to derive continuous-time models. Yet again, the focus lies on the fact that the problem can be discretized in time and the NARX models do not fully exploit the continuous nature of the underlying process.

To devise a less discrete view of ARX modeling, let us first consider a continuous-time signal $f(t)$, as depicted in Figure 1. We set our focus on the signal during a time window $\eta(t^*, T)$, indexed by a time $\tau \in [0, T]$, with endpoint t^* and finite length T, i. e., we define the continuous interval $\eta(t, T) = [t - T, t]$, such that:

$$f(\eta(t^*, T), \tau) = f(t^* - T + \tau). \tag{15}$$

We assume that the signal $f(\eta(t^*, T), \tau)$ exhibits some form of temporal regularity, and that its dynamic information can be represented by a set of continuous temporal features $\boldsymbol{v}(\tau)$, e. g. cosine functions, wavelets, polynomials, etc., of the form:

$$f(\eta(t^*, T), \tau) = \sum_{i=1}^{\infty} \langle f(\eta(t^*, T), \tau), v_i(\tau) \rangle_\tau \, v_i(\tau) = \sum_{i=1}^{\infty} \xi_i(t^*) v_i(\tau), \tag{16}$$

where the $\xi_i(t^*) \stackrel{\text{def}}{=} \langle f(\eta(t^*, T), v(\tau) \rangle_\tau = \langle f(t^* - T + \tau), v(\tau) \rangle_\tau$ represent the projection of the original function in the time window $\eta(t^*, T)$ onto the i-th continuous temporal feature $v_i(\tau)$. An

interesting property of the right-hand side of Eq. (16) is that it explicitly decomposes the time dependence of the underlying function $f(t)$ in two different components: the scalar coefficients $\boldsymbol{\xi}(t^*)$ encoding the "global" time t^*, but not the "local" time τ within the window, while the continuous features v_i encode the dependence on τ instead. In other words, we decompose a moving window of duration T, starting from the global time $t^* - T$, on a set of temporal features $v_i(\tau)$.

For illustrative purposes, the bottom plot in Figure 1 illustrates such a finite set of continuous features $v_1(\tau), \ldots, v_{N_F}(\tau)$ weighted by their coefficients $\xi_1(t^*), \ldots, \xi_{N_F}(t^*)$, such that the response of the function $f(t)$ on the highlighted time window $\eta(t^*, T)$ is decomposed as:

$$f(\eta(t^*, T), \tau) = \sum_{i=1}^{N_F} \xi_i(t^*) v_i(\tau), \quad \tau \in [0, T]. \tag{17}$$

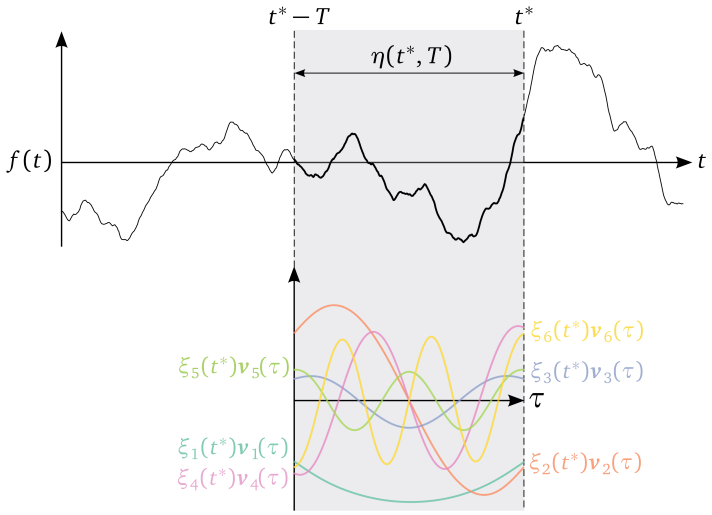


Figure 1: (Top) Continuous-time signal $f(t)$ indexed by the "global" time t with highlighted time window $\eta(t^*, T)$ with length T. (Bottom) Temporal signal features $\boldsymbol{v}_i(\tau)$ indexed by the "local" time τ and scaled by the corresponding coefficients $\xi_i(t^*)$.

Representing a time-dependent signal through its properties within a fixed-width moving window is a well-established technique in signal processing, widely known as the *sliding window* Bastiaans (1985); Badeau et al. (2004). Nevertheless, to the authors' best knowledge, this is the first time this approach is combined within an autoregressive setting. Popular choices to represent time-dependent signals in the form of Eq. (16) include Fourier transforms Bracewell (1989), and Karhunen-Loève expansions Loève (1955). An advantage of structured representations like these is that real-world processes are often represented adequately even using a heavily truncated set of features. This property is taken advantage of, for example, for dimensionality reduction Bengio et al. (2006) or lossy signal compression Strang and Nguyen (1996). Autoregressive inputs and responses often exhibit strong temporal regularity and predictable behavior, originating from the inert physical processes that govern them, and are therefore suitable for this type

of representation. For physical systems, domain-specific knowledge may be even leveraged to produce particularly efficient representations.

In the context of autoregressive simulation, additional features may also be considered to simplify the construction of the input/output map in Eq. (3). These include classical statistics in the sliding window, such as moving averages, variances, dominating frequencies, principal components, singular value decomposition eigenmodes, among others. The usefulness of using prior knowledge about the physics of the system in the ARX modeling process has recently been demonstrated in Schär et al. (2024).

In the following section, we will discuss how we can use features of the exogenous inputs and output to reformulate the classical discrete-time view on ARX into a continuous functional view that can mitigate or even eliminate some of its limitations.

2.4 Functional nonlinear autoregressive with exogenous inputs modeling ($\mathcal{F}$-NARX)

We introduce here our adaptation of NARX modelling designed to work with features, rather than lags. We start by defining the time-dependent *information vector* $\boldsymbol{\zeta}(t)$, that gathers all exogenous inputs together with the model response within a time window $\eta(t, T)$ as follows:

$$\boldsymbol{\zeta}(t) = \{x_1(\eta(t,T)), \dots, x_M(\eta(t,T)), y(\eta(t-\delta t, T-\delta t))\}, \tag{18}$$

where $0 < \delta t \ll T$ is a small positive arbitrary time increment.

Intuitively, the window width T can be considered as the *memory* of the model. Ideally, it is close to the *effective memory* of the system with respect to the corresponding variable. We define the effective memory as the look-back time such that the effect of the independent variable $x_i(t)$ on the dependent variable $y(t+\delta t)$ in Eq. (3) becomes negligibly small. In principle, when considering complex systems with different processes and inputs, a different memory could be considered for each component of $\zeta_i(t)$ in Eq. (18), resulting in a set of memories $\boldsymbol{T} = \left\{T_1, \cdots, T_{|\boldsymbol{\zeta}|}\right\}$. To simplify our notation and avoid adding complexity, we assume, without loss of generality, a uniform memory T for all input and output components from this point onward. For each of the $(M+1)$ components $\zeta_j(t)$ in Eq. (18), we now introduce a corresponding function $\mathcal{K}_j$, which extracts a set of features $\boldsymbol{\xi}_j(t)$:

$$\boldsymbol{\xi}_j(t) = \mathcal{K}_j(\zeta_j(t)). \tag{19}$$

No restrictions are made on the nature of the transforms $\mathcal{K}_j$, nor on the dimensionality of each feature-set $\boldsymbol{\xi}_j(t)$. Different elements of the information vector $\boldsymbol{\zeta}(t)$ can have different transforms, or different truncation schemes. For example, some exogenous input $\zeta_p(t)$ may exhibit a strong cyclic behavior, and the corresponding $\boldsymbol{\xi}_p(t)$ may represent a subset of its Fourier coefficients, while some other input, say $\zeta_q(t)$, may instead have a smoother behavior, better described by a polynomial representation with coefficients $\boldsymbol{\xi}_q(t)$.

We gather all the feature sets $\boldsymbol{\xi}_j$ into a single feature vector $\boldsymbol{\xi}(t)$, with $|\boldsymbol{\xi}(t)| = \tilde{n}$:

$$\boldsymbol{\xi}(t) = \left\{\boldsymbol{\xi}_{x_1}(t), \cdots, \boldsymbol{\xi}_{x_M}(t), \boldsymbol{\xi}_y(t)\right\}. \tag{20}$$

Note that $\tilde{n}$ is the total number of relevant features extracted from the $M+1$ components. We can now adapt the classical NARX approach in Section 2.1, by capitalizing on the feature vector $\boldsymbol{\xi}(t)$. The goal is to approximate a near future response value $\widehat{y}(t+\delta t) \approx y(t+\delta t)$ as a function of $\boldsymbol{\xi}(t)$:

$$\widehat{y}(t+\delta t) = \widehat{\mathcal{M}}(\boldsymbol{\xi}(t+\delta t); \boldsymbol{c}). \tag{21}$$

An interesting aspect of this formulation is that it does not make any assumption about the discretization of the time axis. This is because the features $\boldsymbol{\xi}$ in Eq. (16)) are not directly dependent on the local time axis τ (see Figure 1), only the transform to extract them is. In other words, the specific discretization choices of the sliding time window $\eta(t,T)$ only affect how the features are extracted (and therefore possibly the accuracy of their extraction), but not their interpretation or physical meaning. To give a concrete example, assume that the chosen features of a chosen continuous window $\zeta_p(t^*,\tau)$ can be exactly represented by a second order polynomial. The set of features $\boldsymbol{\xi}(t^*)$ is then given by the three polynomial coefficients $\boldsymbol{\xi}(t^*) = \{\xi_0(t^*), \xi_1(t^*), \xi_2(t^*)\}$ such that:

$$\zeta_p(t^*,\tau) = \xi_0(t^*) + \xi_1(t^*)\tau + \xi_2(t^*)\tau^2, \quad \tau \in [0,T]. \tag{22}$$

If $\zeta_p(t^*,\tau)$ is then arbitrarily discretized on the time axis, the values of $\boldsymbol{\xi}(t^*)$ will remain unchanged. Nevertheless, their extraction accuracy from a discretized dataset may depend on the discretization itself. In the given example, at least three time samples would be needed to exactly determine the coefficient of a second degree polynomial interpolant. This natural robustness to oversampling is especially interesting in the context of surrogate modeling of numerical dynamical systems, where the sampling frequency of the solver is often many times higher than the actual frequency content of the modeled signal responses. It is also an important property for the mitigation of the over-reliance problem of classical NARX at high sampling rates (see Section 2.2).

Note that $\boldsymbol{\zeta}(t)$ in Eq. (18) is essentially a continuous equivalent of $\boldsymbol{\varphi}(t)$ in Eq. (6), and that we can write the time window in a discretized form as follows:

$$\eta(t,T) = \{t, t-\delta t, \ldots, t-n_t\delta t\} \tag{23}$$

where $n_t = \lceil T/\delta t \rceil$. The key difference to the classical ARX modeling introduced in Section 2.1 lies in the use of the features $\boldsymbol{\xi}(t)$ generated through the functions $\mathcal{K}_1, \ldots, \mathcal{K}_{(M+1)}$, rather than in the original set of input/outputs $\{\boldsymbol{x}, \boldsymbol{y}\}$, hence exploiting the smoothness and regularity of the process being modeled.

A direct advantage of this formulation is that all the existing NARX-fitting strategies introduced in Section 2.1 can be adapted with minimal modifications if the problem at hand has a discretized

time axis. However, Eqs. (21) and (18) also apply for continuous-time systems, and they do not leverage or depend on any specific discretization choice. Because in this setting, the ARX model relies on the functional features of the inputs and outputs within a sliding window, rather than individual time steps, we refer to our approach as *functional nonlinear autoregressive with exogenous inputs* ($\mathcal{F}$-NARX) modeling.

Because the overwhelming majority of numerical modeling of dynamic systems relies on numerical computations, which are inherently discrete, in the remainder of this work we will focus on discrete-time systems and how these applications can benefit from $\mathcal{F}$-NARX models. We start by observing that many of the feature extraction methods for continuous-time signals introduced in Section 2.3 admit a discrete counterpart. Classical examples include the discrete version of the Fourier transform Sundararajan (2001), or principal component analysis Pearson (1901) as the discrete alternative to the Karhunen-Loève expansion. Other popular methods include auto-encoders Rumelhart and McClelland (1986), wavelet transforms Olkkonen (2011) and Isomaps Tenenbaum et al. (2000). These methods can efficiently represent the memory of each variable, improving the scalability of $\mathcal{F}$-NARX models with respect to the effective system memories.

2.5 $\mathcal{F}$-NARX model fitting and prediction

We recall from Section 2.1 that to fit a classical NARX model, for each observation (i) from the experimental design, we need to construct the design matrix $\boldsymbol{\Phi}^{(i)} \in \mathbb{R}^{\widetilde{N}\times n}$ and the corresponding output vector $\boldsymbol{y}^{(i)} \in \mathbb{R}^{\widetilde{N}}$ as defined in Eq. (7). A similar procedure can be followed with the $\mathcal{F}$-NARX model. Given an observation (i), we keep our definition of the vector $\boldsymbol{y}^{(i)}$ but we also define the feature matrix $\boldsymbol{\Xi}^{(i)} \in \mathbb{R}^{\widetilde{N}\times \tilde{n}}$ as an equivalent to the matrix $\boldsymbol{\Phi}^{(i)}$:

$$\boldsymbol{\Xi}^{(i)} = \begin{pmatrix} \boldsymbol{\xi}^{(i)}(t_0) \\ \boldsymbol{\xi}^{(i)}(t_0+\delta t) \\ \vdots \\ \boldsymbol{\xi}^{(i)}(t_0+(N-1)\delta t) \end{pmatrix}. \tag{24}$$

In analogy to Eq. (8), the regression task can then be performed on the input matrix $\boldsymbol{\Xi}^{\mathrm{ED}}$ and the output vector $\boldsymbol{y}^{\mathrm{ED}}$ comprising the data of the full experimental design:

$$\boldsymbol{\Xi}_{\mathrm{ED}} = \begin{pmatrix} \boldsymbol{\Xi}^{(1)} \\ \vdots \\ \boldsymbol{\Xi}^{(N_{\mathrm{ED}})} \end{pmatrix}, \quad \boldsymbol{y}_{\mathrm{ED}} = \begin{pmatrix} \boldsymbol{y}^{(1)} \\ \vdots \\ \boldsymbol{y}^{(N_{\mathrm{ED}})} \end{pmatrix}. \tag{25}$$

Ending up with this classical regression setting as is useful, for example, when the duration of individual realizations is long, or the experimental design is large. In these cases $\boldsymbol{\Xi}^{\mathrm{ED}}$ becomes very large and efficient regression fitting using e. g. ordinary least squares as shown in Eq. (11) can limited by the available computing resources, memory in particular. In such a scenario, a viable solution is to perform the regression on a subset of the sample pairs instead of using the

full matrix $\boldsymbol{\Xi}_{\mathrm{ED}}$ and vector $\boldsymbol{y}_{\mathrm{ED}}$. This subsampling approach has been recently discussed and exploited in Schär et al. (2024) and will also be used in the applications Section 4.

During the prediction phase, the $\mathcal{F}$-NARX approach is similar to classical NARX modeling, in that it uses its own past predictions to predict new output time steps:

$$\widehat{y}(t+\delta t) = \widehat{\mathcal{M}}(\widehat{\boldsymbol{\xi}}(t+\delta t)), \tag{26}$$

where $\widehat{\boldsymbol{\xi}}(t) = \left\{\boldsymbol{\xi}_{x_1}(t), \cdots, \boldsymbol{\xi}_{x_M}(t), \widehat{\boldsymbol{\xi}}_y(t)\right\}$ is the feature equivalent to the vector $\widehat{\boldsymbol{\varphi}}(t)$ in Eq. (14), built using the previously predicted outputs:

$$\widehat{\boldsymbol{\xi}}_y(t) = \mathcal{K}_y(\widehat{\boldsymbol{\zeta}}_y(t)) \quad \text{with} \quad \widehat{\boldsymbol{\zeta}}_y(t) = \widehat{y}(\eta(t-\delta t, T-\delta t)). \tag{27}$$

Note the additional feature extraction step $\mathcal{K}_j$, which has to be performed at each prediction time step and for each input j. If a very high forecast rate is required, the transform $\mathcal{K}_j$ should therefore be computationally efficient to not compromise the forecast performance of the final surrogate. It is worth noting, however, that this additional cost can be compensated for during subsequent processing of the vector $\widehat{\boldsymbol{\xi}}(t)$. Consider the classical NARX prediction as shown in Eq. (13) and (14). The mapping $\mathcal{G}$ is typically computationally more expensive if $\widehat{\boldsymbol{\varphi}}(t)$ is large, which is e. g. the case if the system memory and thus the model orders are large. The feature extraction step can produce a $\widehat{\boldsymbol{\xi}}(t)$ that is considerably smaller than $\widehat{\boldsymbol{\varphi}}(t)$ in the classical NARX setting, thus speeding up the evaluation of $\mathcal{G}$.

3 Sparse $\mathcal{F}$-NARX modeling using principal component analysis and polynomial regression

After introducing the general $\mathcal{F}$-NARX modeling approach in Sections 2.3-2.5, we now establish a concrete implementation for applications following in Section 4. This implementation utilizes principal component analysis (PCA) for the calculation of the temporal features and it additionally uses polynomial basis functions to introduce nonlinearity in the ARX prediction.

3.1 Principal component analysis

Let us consider a discretized version of the information matrix $\boldsymbol{\zeta} \in \mathbb{R}^{\widetilde{N} \times n}$ with rows $\boldsymbol{\zeta}(t) \in \mathbb{R}^n$ as introduced in Eq. (18) using discrete time windows as defined in Eq. (23). Our goal is to apply a transform $\mathcal{K}_i$ to each element $\boldsymbol{\zeta}_i \in \mathbb{R}^{\widetilde{N} \times n_i}$ of $\boldsymbol{\zeta}$ to obtain a discrete feature matrix $\boldsymbol{\Xi}_i \in \mathbb{R}^{\widetilde{N} \times \tilde{n}_i} = \mathcal{K}_i(\boldsymbol{\zeta}_i)$. A well known tool to extract relevant features from auto-correlated discrete signals is given by principal component analysis (PCA) Jolliffe (2002), which can be written as:

$$\boldsymbol{\Xi}_i = \mathcal{K}_i^{\mathrm{PCA}}(\boldsymbol{\zeta}_i) = \boldsymbol{\zeta}_i \boldsymbol{\Lambda}_i. \tag{28}$$

The transformation matrix $\boldsymbol{\Lambda}_i$ is obtained by first standardizing $\boldsymbol{\zeta}_i$ to have zero mean and unit variance:

$$\boldsymbol{Z}_i = \frac{\boldsymbol{\zeta}_i - \mu_i}{\sigma_i}, \tag{29}$$

where $\mu_i \in \mathbb{R}^{n_i}$ and $\sigma_i \in \mathbb{R}^{n_i}$ are the sample means and standard deviations of $\boldsymbol{\zeta}_i$, respectively. Subsequently, the covariance matrix $\boldsymbol{C}_i \in \mathbb{R}^{n_i \times n_i}$ is computed as:

$$\boldsymbol{C}_i = \frac{1}{\widetilde{N} - 1} \boldsymbol{Z}_i^\top \boldsymbol{Z}_i, \tag{30}$$

and an eigenvalue decomposition is performed to obtain its eigenvalues $\lambda_{ij} \in \mathbb{R}$ and eigenvectors $\boldsymbol{v}_{ij} \in \mathbb{R}^{n_i}$:

$$\boldsymbol{C}_i \boldsymbol{v}_{ij} = \lambda_{ij} \boldsymbol{v}_{ij}. \tag{31}$$

The matrix $\boldsymbol{\Lambda}_i$ then gathers the $\tilde{n}_i$ eigenvectors corresponding to the largest eigenvalues, in decreasing order of magnitude:

$$\boldsymbol{\Lambda}_i = \{\boldsymbol{v}_{i1}, \boldsymbol{v}_{i2}, \ldots, \boldsymbol{v}_{i\tilde{n}_i}\}. \tag{32}$$

In practical applications, $\boldsymbol{\Lambda}_i$ is usually truncated to a much smaller subset, corresponding to the largest $\tilde{n}_i$ principal components. Consequently, the transform $\mathcal{K}_i^{\text{PCA}}$ can be parametrized in terms of the number of principal components $\tilde{n}_i$ where $1 \leq \tilde{n}_i \leq n_i$. A more intuitive approach to the parameterization is to adopt the *explained variance* ν_i, which corresponds to the fraction of the total variance of the signal that is reconstructed by the chosen truncated set, which can be computed as:

$$\nu_i = \frac{\sum_{k=1}^{\tilde{n}_i} \lambda_{ik}}{\sum_{\ell=1}^{n_i} \lambda_{i\ell}}. \tag{33}$$

By projecting each information vector $\boldsymbol{\zeta}_i(t)$ onto its corresponding PCA basis, we introduce a natural ranking in the features, as $\boldsymbol{v}_{i1}$ explains most of the variance of the i-th variable, followed by $\boldsymbol{v}_{i2}$, etc. This is different from classical ARX modeling, where such an order does not exist. For example, the shortest lags are not necessarily the most important ones, one of the reasons why the selection of lags is a well-known challenge in autoregressive modeling, as discussed in Section 2.2. However, it should be noted that while $\boldsymbol{v}_{i1}$ explains most of the signal variance, this does not guarantee that it is also the most important mode for predicting the system response, which can also be governed by higher modes instead. Nevertheless, the parametrization of the $\mathcal{F}$-NARX model in terms of explained variance instead of discrete time lags can be considered an easier parametrization since a change in ν_i has more predictable consequences on the model performance compared to including or excluding individual lags.

3.2 Polynomial $\mathcal{F}$-NARX model

Recall from Section 2.1 that NARX models are often formulated as linear-in-the-parameters models, leading to a linear regression problem as defined in Eq. (10), where a popular choice for the nonlinear mapping $\mathcal{G}$ is polynomials. Polynomials have a long tradition in the use of NARX

models and have proven to perform well in a variety of problems Billings (2013). In addition, they are easy to parameterize and computationally efficient. In this section, we will explain how polynomials can be applied to build a polynomial $\mathcal{F}$-NARX model.

Given any discrete feature vector $\boldsymbol{\xi}(t) \in \mathbb{R}^{\tilde{n}}$ from the feature matrix $\boldsymbol{\Xi} \in \mathbb{R}^{\widetilde{N} \times \tilde{n}}$, we can construct the monomial $\mathcal{P}_{\boldsymbol{\alpha}}(\boldsymbol{\xi}(t))$ as follows:

$$\mathcal{P}_{\boldsymbol{\alpha}}(\boldsymbol{\xi}(t)) = \prod_{i=1}^{\tilde{n}} \xi_i(t)^{\alpha_i}, \tag{34}$$

where $\xi_i(t)$ is the i-th component of $\boldsymbol{\xi}(t)$ and $\boldsymbol{\alpha} \in \mathbb{N}^{\tilde{n}}$ is an integer multi-index. This allows us to approximate the output $y(t)$ as a weighted sum of monomials, where the weights $c_{\boldsymbol{\alpha}}$ are a set of real-valued coefficients:

$$y(t) = \sum_{\boldsymbol{\alpha} \in \mathcal{A}} c_{\boldsymbol{\alpha}} \mathcal{P}_{\boldsymbol{\alpha}}(\boldsymbol{\xi}(t)). \tag{35}$$

The multi-index set $\mathcal{A}$ is truncated to control the model complexity. In this study, we will use a truncation strategy that relies on three parameters: total polynomial degree d, maximum allowed interaction r, and hyperbolic truncation index q as introduced in the context of polynomial chaos expansions by Blatman and Sudret (2010). Consequently, the multi-index $\boldsymbol{\alpha} \in \mathcal{A}^{\tilde{n},d,r,q}$ is constrained as follows:

$$\mathcal{A}^{\tilde{n},d,r,q} = \left\{ \boldsymbol{\alpha} \mid (||\boldsymbol{\alpha}||_0 \leq r) \cap (||\boldsymbol{\alpha}||_q \leq d) \right\}, \tag{36}$$

where $||\boldsymbol{\alpha}||_0 = \sum_{i=1}^{\tilde{n}} \mathbb{1}_{\{\alpha_i > 0\}}$ and $||\boldsymbol{\alpha}||_q = \left(\sum_{i=1}^{\tilde{n}} \alpha_i^q \right)^{1/q}$ for $0 < q \leq 1$.

Identifying the model coefficients $c_{\boldsymbol{\alpha}}$ can then be solved as an ordinary regression problem using e. g. ordinary least squares as shown in Eq. (11) with regression matrix $\boldsymbol{\Psi} \in \mathbb{R}^{\widetilde{N} \times p}$ and output vector $\boldsymbol{y} \in \mathbb{R}^{\widetilde{N}}$:

$$\boldsymbol{\Psi} = \begin{pmatrix} \boldsymbol{\mathcal{P}}(t_0) \\ \vdots \\ \boldsymbol{\mathcal{P}}(t_{\max}) \end{pmatrix}, \quad \boldsymbol{y} = \begin{pmatrix} y(t_0) \\ \vdots \\ y(t_{\max}) \end{pmatrix}. \tag{37}$$

Here, we use $\boldsymbol{\mathcal{P}}(t)$ as shorthand notation for $\mathcal{P}_{\boldsymbol{\alpha}}(\boldsymbol{\xi}(t))$.

3.3 Hybrid least-angle regression

To solve the regression problem introduced in the previous section, we adopt least angle regression (LARS), a well-known sparse regression approach from the compressive sensing literature. Least angle regression, first introduced in Efron et al. (2004), is a stepwise linear regression technique often used when the number of regressors is much larger than the number of samples in the regression problem. Due to the use of lagged inputs and outputs, classical NARX models can fall into this category even when the dimensionality of the exogenous inputs is relatively low. LARS has been successfully employed in the recent literature for the construction of sparse NARX models Mai et al. (2016); Bhattacharyya et al. (2020); Li et al. (2021). Although $\mathcal{F}$-NARX

modeling can reduce the dimensionality compared to classical NARX modeling, using a sparse solver can still be beneficial, especially when considering problems with high exogenous-input dimensionality, high polynomial degrees or high interaction orders (see Section 3.2).

The LARS algorithm, as presented in Efron et al. (2004), comprises the following steps:

1. **Initialization**: Let $\boldsymbol{c}^{(k)} \in \mathbb{R}^p$ be the coefficient vector at iteration k and initialize it as a vector of all zeros: $\boldsymbol{c}^{(0)} = \boldsymbol{0}$.

2. **Calculate residuals**: Compute the residual vector $\boldsymbol{y} - \boldsymbol{\Psi}_z \boldsymbol{c}^{(k)}$ where $\boldsymbol{\Psi}_z \in \mathbb{R}^{\widetilde{N} \times p}$ is the regression matrix as in Eq. (37) standardized to have zero mean and unit variance, and $\boldsymbol{y} \in \mathbb{R}^{\widetilde{N}}$ is the corresponding output vector.

3. **Find most correlated regressor**: Identify the regressor most correlated with the residuals and add it to the active set of regressors. We gather the indices of the active regressors in $\boldsymbol{\mathcal{A}}$.

4. **Update coefficients**: Equally increase all the coefficients of the active regressors until one inactive regressor achieves the same correlation with the residuals as the current ones.

5. **Iterate**: Repeat steps 2 to 4 until all regressors are selected or a stopping criterion is met. In our work, we stop the algorithm when a given number of non-zero coefficients is reached. This number is a parameter of the $\mathcal{F}$-NARX algorithm.

The LARS algorithm generates a sequence of coefficient vectors, often called the *LARS path*, with each vector corresponding to one iteration. These vectors contain an increasing number of non-zero coefficients, as LARS adds one regressor per iteration, without removing any. It is well known that the LARS-generated coefficients on the normalized and scaled data do not provide optimal prediction accuracy, and in their original work Efron et al. (2004) introduced the idea of *hybrid LARS-OLS*. Instead of directly using the coefficients estimated by LARS at any given point of its path, the ordinary least-squares solution can be computed on the original non-standardized data and regressors as follows:

$$\widehat{\boldsymbol{c}}^{(k)} = \underset{\boldsymbol{c}_{\boldsymbol{\mathcal{A}}}^{(k)}}{\arg\min} \|\boldsymbol{y} - \boldsymbol{\Psi} \boldsymbol{c}^{(k)}\|^2. \tag{38}$$

Here, $\boldsymbol{c}_{\boldsymbol{\mathcal{A}}}^{(k)}$ are the coefficients corresponding to the regressors selected by LARS until iteration k, and their values are computed using ordinary least squares.

It is important to note that the solutions $\widehat{\boldsymbol{c}}^{(k)}$ minimize the one-step-ahead (OSA) prediction error. However, a small OSA prediction error does not necessarily indicate good forecast performance. In the literature, various methods have been developed to reduce the forecast error. These typically involve pruning, the process of selecting a single regressor to be eliminated at each iteration Piroddi and Spinelli (2003); Piroddi (2008); Piroddi et al. (2010), which can be computationally

extremely expensive. Alternatively, gradient-based methods have been proposed, but they also incur high computational costs Farina and Piroddi (2009, 2010).

To obtain a $\mathcal{F}$-NARX model with good forecast performance, while keeping the model fitting costs manageable, we leverage the LARS path to enhance the forecast performance. For a coefficient vector $\widehat{\boldsymbol{c}}^{(k)}$, we compute the mean forecast error over the experimental design using the normalized mean squared error:

$$\varepsilon_y^{(i)}(\widehat{\boldsymbol{c}}^{(k)}) = \frac{1}{N_t} \frac{\sum_{t=t_0}^{t_{\max}} \left[y^{(i)}(t) - \widehat{y}^{(i)}(\widehat{\boldsymbol{c}}^{(k)}, t) \right]^2}{\mathrm{Var}(\boldsymbol{y}^{(i)}) + \gamma}, \tag{39}$$

where $\boldsymbol{y}^{(i)}$ is the output of the i-th realization in the experimental design, $\mathrm{Var}(\boldsymbol{y}^{(i)})$ its variance, $\widehat{\boldsymbol{y}}^{(i)}$ is the corresponding model forecast, and $N_t = (t_{\max} - t_0)/\delta t + 1$ is the number of forecasted time steps. The variable γ is a small positive regularization term added to the denominator to avoid division by zero if a signal is constant. It can also prevent giving excessively high weight to signals with very low variance compared to the rest of the signals in the experimental design. The mean forecast error value of interest is then computed as:

$$\overline{\varepsilon_y}(\widehat{\boldsymbol{c}}^{(k)}) = \frac{1}{N_{\mathrm{ED}}} \sum_{i=1}^{N_{\mathrm{ED}}} \varepsilon_y^{(i)}(\widehat{\boldsymbol{c}}^{(k)}), \tag{40}$$

and we select the coefficient vector that minimizes this mean forecast error over all LARS iterations as our final model coefficients:

$$\widehat{\boldsymbol{c}} = \arg\min_k \overline{\varepsilon_y}(\widehat{\boldsymbol{c}}^{(k)}). \tag{41}$$

This approach not only leads to better prediction performance, but it also reduces the risk of overfitting, as the coefficient vector is selected using a different metric than the one minimized in Eq. (38).

Evaluating the forecast error for all coefficient vectors produced by LARS can be computationally demanding, as the number of vectors can reach $\mathcal{O}(10^{2-3})$ for complex systems requiring many active regressors. In these cases, not evaluating all coefficient vectors $\widehat{\boldsymbol{c}}^{(k)}$ can be the preferred approach. For example, sets with very few non-zero coefficients are unlikely to be sufficient to describe very complex systems. The fitting costs can be further reduced by only evaluating every k-th LARS iteration instead of evaluating after every iteration.

It is worth noting the similarity of this approach to the work of Blatman and Sudret (2011), which used LARS with a cross-validation scheme to select the best set of regressors for time-independent problems. The approach presented above can therefore be seen as an adaptation of their work to time-dependent problems.

4 Applications

In this section we present the application of $\mathcal{F}$-NARX modeling to two case studies, namely:

- a tall building under wind loading in Section 4.1, to demonstrate the robustness of the algorithm to changes in parameters such as the size of the training set, the sampling rate of the problem, and the memory window length;
- a three-story steel frame under seismic loading in Section 4.2, to demonstrate the suitability of $\mathcal{F}$-NARX to emulate the extreme behavior of complex dynamic system for, e. g. reliability and fragility analysis.

4.1 Eight-story building under wind loading

In our first case study, we consider the eight-story building displayed in Figure 2, which we adopted from Kim et al. (2023). The building is modeled as a linear elastic lumped mass system with the system parameters listed in Table 1. The structure is subject to dynamic wind forces. The wind field generating these forces is modeled as a temporal random field, as described in Kim et al. (2023). At a given time instant, the wind field consists of eight wind speed values corresponding to each of the eight building nodes shown in the right panel of Figure 2. The parameters of the wind field model are mostly identical to those listed in Kim et al. (2023). The only exception is the basic wind speed V_b, which we model as a random variable following a normal distribution with a mean of 42.5 m/s and a coefficient of variation of 33.3 %, in contrast to the deterministic value of 42.5 m/s used in Kim et al. (2023). This choice is motivated by a need to increase the wind field complexity and variability to be closer to a realistic scenario.

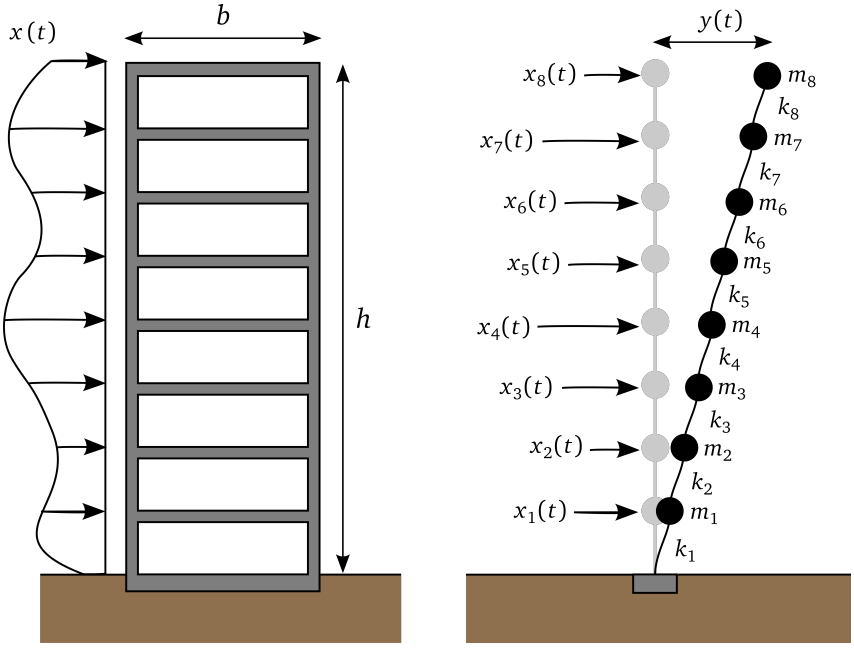


Figure 2: (Left) Sketch of the eight-story building model with horizontal wind loading $x(t)$. (Right) Corresponding lumped mass system with discretized wind loads $x_i(t)$(adapted from Kim et al. (2023)).

Table 1: Eight-story building – System parameters

Parameter	Unit	Value
Height h	m	32.3
Width b	m	36.6
Weight $m_1, \ldots, m_8$	kg	$9.66 \cdot 10^6$
Stiffness $k_1, \ldots, k_8$	N/m	$1.09 \cdot 10^9$
Damping ratio $c_1, \ldots, c_8$	-	0.02

In this case study we assess the ability of $\mathcal{F}$-NARX to predict the horizontal top floor displacement $y(t)$ of the building. The exogenous model input is the wind field, comprising the eight wind speed time series $x_1(t), \ldots, x_8(t)$. We investigate the performance of $\mathcal{F}$-NARX for different configurations, and assess its robustness to the time discretization of the problem. We also compare the performance of $\mathcal{F}$-NARX to a classical NARX model. To perform this analyses, we use a set of 2,100 system realizations, each with a duration of 10 minutes and a sampling rate of 40 Hz. The simulations were conducted using the MATLAB codes provided by the authors of Kim et al. (2023). A randomly selected set of 100 of these simulations will serve as the experimental design, and the remaining 2,000 simulations will be used as a hold-out validation dataset.

4.1.1 Autoregressive model configurations

A total of 17 $\mathcal{F}$-NARX models, using the configurations listed in Table 2, were constructed. The models are parameterized in terms of the explained variances ν_i and the model memories T_i. Note that the explained variances and memories were chosen homogeneously between the exogenous inputs and the autoregressive input. For simplicity, we will therefore refer to them as ν and T, omitting the subscript. We also vary the experimental design size N_{ED} and the time discretization δt of the problem. It is worth noting that the smaller experimental designs are a subset of the largest one and sampling rates other than 40 Hz were not simulated but simply obtained by up-sampling or decimation of the 40 Hz signals.

For Models 1-4, we keep ν, T and δt fixed and only vary the experimental design size N_{ED} from $1 - 100$ simulations. Models 5-10 have fixed N_{ED}, T and δt, but we vary the explained variance ν from $0.85 - 0.995$. For Models 11-13, N_{ED}, ν and δt are fixed, while the model memory T is varied from $1.0 - 4.0$ s. Finally, for Models 14-17, we fix N_{ED}, ν and T and vary the time discretization δt from 0.1 s down to 0.006 s.

For each of the 17 $\mathcal{F}$-NARX models we followed an adaptive approach to select the polynomial basis functions. We tested polynomial degrees (d) from 1 to 3, interaction orders (r) from 1 to 3, and q-norms of 0.7, 0.85, and 1.0. The polynomial coefficients were calculated using hybrid LARS as described in Section 3.3. We ran the LARS algorithm for a maximum of 200 iterations

Table 2: Eight-story building – $\mathcal{F}$-NARX model configurations

	N_{ED} [-]	ν [-]	T [s]	δt [s]
Model 1	1	0.950	1.0	0.025
Model 2	10	0.950	1.0	0.025
Model 3	40	0.950	1.0	0.025
Model 4	100	0.950	1.0	0.025
Model 5	100	0.850	1.0	0.025
Model 6	100	0.900	1.0	0.025
Model 7	100	0.930	1.0	0.025
Model 8	100	0.970	1.0	0.025
Model 9	100	0.990	1.0	0.025
Model 10	100	0.995	1.0	0.025
Model 11	100	0.950	0.5	0.025
Model 12	100	0.950	2.0	0.025
Model 13	100	0.950	4.0	0.025
Model 14	100	0.950	1.0	0.100
Model 15	100	0.950	1.0	0.050
Model 16	100	0.950	1.0	0.0125
Model 17	100	0.950	1.0	0.00625

and we evaluated the forecast performance on the experimental design for every 10-th LARS iteration to speed up the model fitting process. To reduce computational costs, we used only a subset of at most 120,000 rows from the design matrix to construct the surrogates, as described in Section 2.5. This maximum was exhausted for all experimental design sizes except $N_{\mathrm{ED}} = 1$, which comprises only a total of about 24,000 rows. For all 17 models, the adaptive approach selected a setting of $d = 1$, $r = 1$ and $q = 1$. Consequently, the algorithm identified a linear model to best describe the building response. Please note that the algorithm is not limited to linear models, (see, e.g. the application discussed in Section 4.2), but it automatic identified a linear model in this application as the best performing configuration.

In addition to the $\mathcal{F}$-NARX models, we also built a classical NARX model for comparison. For this reference NARX model we used a model order of 15 for both the exogenous inputs and the autoregressive component (see Eq. (6)), and we used the largest experimental design with $N_{\mathrm{ED}} = 100$ simulations for the model fitting. To train the classical NARX model we followed the same basis adaptive approach with the hybrid LARS algorithm as used for the $\mathcal{F}$-NARX models. Also for the reference model, a linear model was found to fit the data best, as expected.

4.1.2 Results

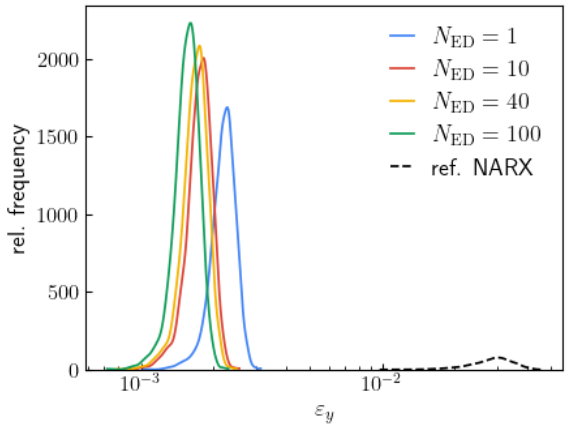
The performance of the $\mathcal{F}$-NARX models and the reference NARX model on the eight-story building under wind load are visualized in Figure 3. To quantify the goodness of the models we calculate the normalized mean-squared error as in Eq. (39) with $\gamma = 0$, and subsequently denoted as ε_y, for every trace in the validation dataset. In Figure 3a, we show the kernel density estimate the distribution of all the ε_y over the validation dataset for four different experimental design sizes N_{ED}. Consequently, the colored curves in this subplot correspond to the Models 1-4 from Table 2. We also show in black the error distribution of the reference NARX model for comparison. Sections of the output traces corresponding to the best and worst prediction of Model 4 (out of 2,000 validation curves) are shown Figure 4 for illustration purposes. In Figure 3a it can be seen that in general $\mathcal{F}$-NARX tends to result in more than one order of magnitude error reduction w. r. t. the reference NARX model. It is also clear that $\mathcal{F}$-NARX models show a clear decreasing error trend with increasing experimental designs. Interestingly, even a single trace in the experimental design can yield relatively good approximation error, at the cost of a much larger trace-to-trace accuracy variability.

Figure 3b shows the $\mathcal{F}$-NARX performance for different values of the explained variance ν (see Eq. (33)). These models correspond to Models 4-9 in Table 2. It can be seen that a ν smaller or equal to 0.93 results in a clear loss in model performance. Very good performance is achieved for $0.95 \leq \nu \leq 0.99$ and the model performs best for $\nu = 0.97$, showing that performance can be compromised if ν is chosen too large. This can be explained by the increased dimensionality in the regression step for large ν, which can still lead to overfitting, as we will discuss in more detail later. For small ν the model cannot capture enough information to approximate the system response.

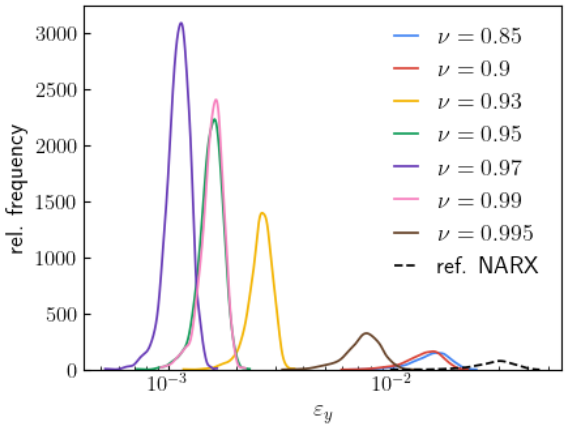
The results for different memories T are given in Figure 3c. The models in this figure correspond to Model 4 and Models 11 to 13 in Table 2. Similarly to the previous case, too small and too high values result in reduced performance, with best performance achieved for $T = 2.0$ s which is about twice the first mode period of the building which is 1.13 s. Analogous to the previous case, this phenomenon may result from the limited information available for very short memories and the large dimensionality and potential overfitting associated for very long memories.

Finally, Figure 3d showcases the behavior of $\mathcal{F}$-NARX different sampling rates of the simulations, compared to the original sampling interval $\delta t = 0.025$ s. These correspond to Models 4 and 14 to 17 in Table 2. It is apparent that upsampling of the data does not affect the model performance. However, if the data is downsampled excessively, the performance starts decreasing. This is expected as high-frequency information in the data necessary to describe the system response can be lost.

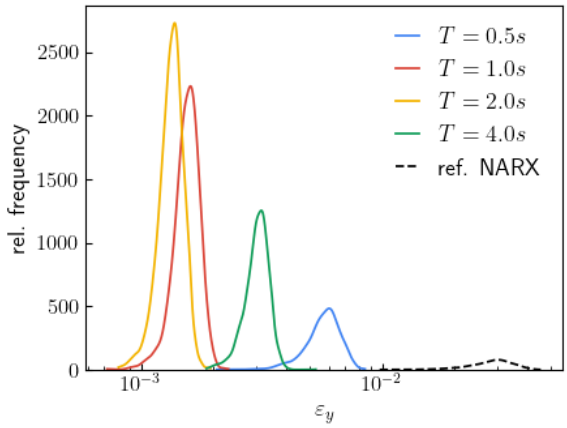
For a more thorough interpretation of the results from Figure 3, we plot the convergence of the explained variance ν_i for an increasing number of principal components n_c and for each of the $\boldsymbol{\zeta}$ components (see Eq. (18)) in Figure 5a. Note that this plot corresponds to a memory



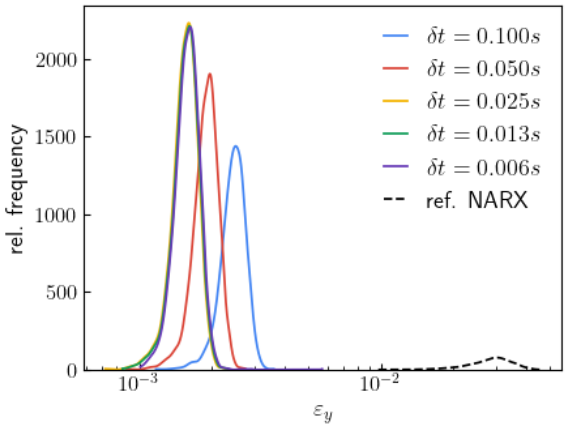
(a) Prediction errors ε_y as in Eq. (39) on the validation dataset for different experiment design sizes N_{ED}. The $\mathcal{F}$-NARX models correspond to Models 1-4 in Table 2.



(b) Prediction errors ε_y as in Eq. (39) on the validation dataset for different explained variances ν. The $\mathcal{F}$-NARX models correspond to Models 4-10 in Table 2.



(c) Prediction errors ε_y as in Eq. (39) on the validation dataset for different memories T. The $\mathcal{F}$-NARX models correspond to the Models 4 and 11-13 in Table 2.



(d) Prediction errors ε_y as in Eq. (39) on the validation dataset for different time increments δt. The $\mathcal{F}$-NARX models correspond to Models 4 and 14-17 in Table 2.

Figure 3: Eight-story building results. (a) Kernel density estimates of the distribution of the prediction errors ε_y (see Eq. (39)) on the validation dataset (2,000 curves) for different numbers of training simulations used to train the $\mathcal{F}$-NARX models (colored lines). The other configuration values were fixed to $\nu = 0.95$ and $T = 1.0$ s and a time increment of 0.025 s was used. The dashed black line shows the distribution of the prediction error for the classical NARX model (see Section 4.1.1) for reference. Note that the abscissa is in log-scale. (b) Distribution of the prediction error for different values of the explained variance. All models were trained on 100 simulations, the memory was fixed to $T = 1.0$ s and the data was sampled with $\delta t = 0.025$ s. (c) Prediction error for different model memories. All models were trained on 100 simulations with the original time increment of $\delta t = 0.025$ s. The explained variance was set to $\nu = 0.95$ for all models. (d) Prediction error for different resampling schemes (i. e., up-sampling or down-sampling). The original time increment is $\delta t = 0.025$ s. All models were trained on 100 simulations and the other parameters were set to $\nu = 0.95$ and $T = 1.0$ s.

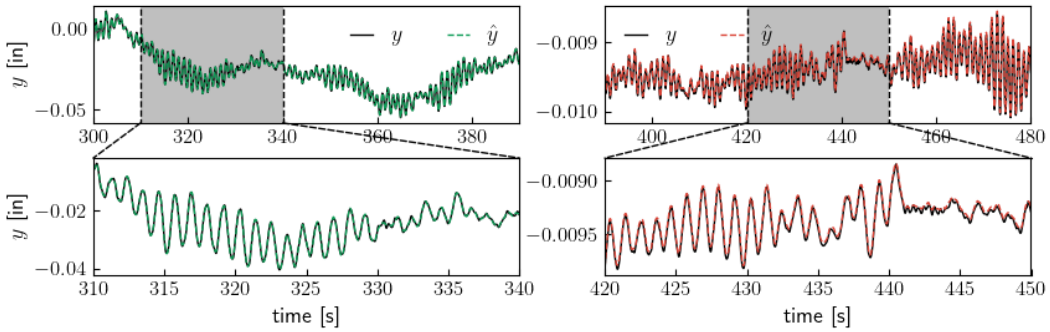
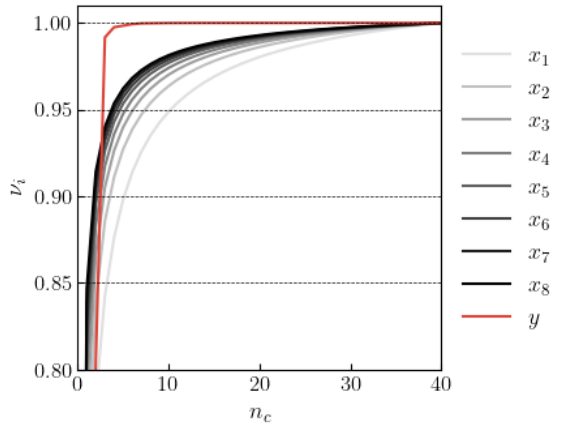


Figure 4: Eight-story building results. (Left) Example traces showing the prediction of model 4 (see Table 2) with the lowest prediction error out of the 2,000 validation curves. The colored line shows the model prediction while the black line shows the simulated output. Note that only a 90-second segment of the 10-minute-long simulation is shown. (Right) True vs. predicted trace with the highest prediction error out of the 2,000 validation curves.

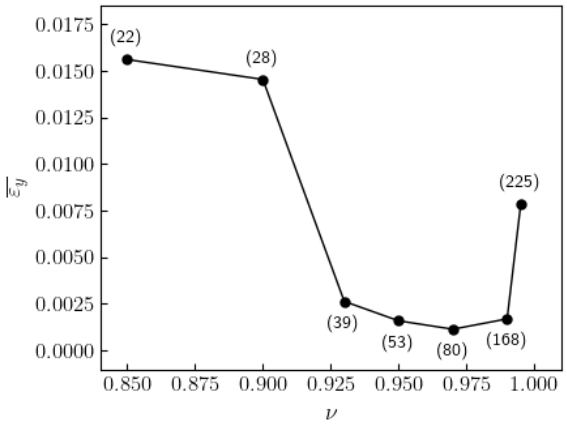
of $T = 1.0$ s with a sampling rate of 40 Hz as e. g. used for Model 4 in Table 2. We observe a fast initial increase in ν_i as n_c gets larger. Consequently, most of the variance in the data can be explained by just a few components. The dimensionality during regression can thus be reduced significantly without much loss of information. As expected, the explained variances for $y, x_1, \ldots, x_8$ converge at different rates, with the model response y converging the fastest. Since ν_i was chosen homogeneously, variables that show slow convergence contribute more to the total dimensionality because they require more components to reach the target explained variance. It can therefore be beneficial to cover less variance of variables with slow convergence if they are found to be less important to the model response. An optimization of this trade-off could be performed through the DRSM approach introduced in Lataniotis et al. (2020), which performs a joint fit of the dimensionality reduction algorithm and surrogate model to improve the final surrogate accuracy.

In Figure 5b we show the evolution of the average ε_y (average over the 2,000 test simulations denoted as $\overline{\varepsilon_y}$) for increasing ν. For each data point, we annotate the corresponding number of regression coefficients, because it is expected to increase significantly with the explained variance ν. It can be observed that as ν increases, the model error first decreases with a large drop from $\nu = 0.9$ to $\nu = 0.93$. This can be explained by the fact that the first principal components explain a lot of variance and these components may also be the most important ones to explain the response dynamics without increasing the dimensionality of the problem much (see Figure 5b). As sufficient information is available to explain the response dynamics, the error plateaus until it increases again at about $\nu = 0.99$. When ν approaches one, only comparatively little information is added by the extra input features, while the dimensionality increases rapidly and causes the model fitting to become underdetermined.

Note that the best value for ν for this problem lies within a relatively small window between 0.925



(a) Number of principal components n_c vs. explained variance ν_i



(b) Mean prediction error $\overline{\varepsilon_y}$ for different values of ν

Figure 5: Eight-story building results. (Left) Convergence of the explained variances ν_i with respect to the number of principal components n_c for each of the eight exogenous inputs x_i and the output quantity y. (Right) Evolution of the mean prediction error $\overline{\varepsilon_y}$ (see Eq. (40)) for increasing values of ν. The corresponding number of model coefficients is annotated in parentheses.

and 0.975. However, choosing a value from this range appears natural due to the interpretability of this value. Alternatively, a plot such as shown in Figure 5b can help in choosing an appropriate value.

4.2 Three-story steel frame under seismic loading

In this application, we showcase the performance of $\mathcal{F}$-NARX on a realistic case study of a nonlinear three-story steel frame under seismic loading, which we adopted from Zhu et al. (2023). The frame structure is illustrated in Figure 6. Geometry parameters, basic material properties and the live load are listed in Table 3. For a detailed explanation of the model we refer to Zhu et al. (2023).

The quantities of interest are the three interstory drifts $\Delta_1(t), \Delta_2(t)$ and $\Delta_3(t)$ (see Figure 6). The exogenous inputs are the ground acceleration $\ddot{x}(t)$, ground velocity $\dot{x}(t)$ and the ground displacement $x(t)$ caused by seismic events. The ground acceleration data is taken from the Pacific Earthquake Engineering Research Center (PEER) database Power et al. (2008) which gathers recordings of earthquake data near the earth's surface or on structure. In this case study we select earthquakes that leave the structure without significant permanent damage, as plastic deformation and damage accumulation would essentially require an infinite memory window. In turn, this behavior would make the problem unsuitable for a straightforward application of autoregressive modeling, which assumes finite system memory. Recent advancements on dynamic

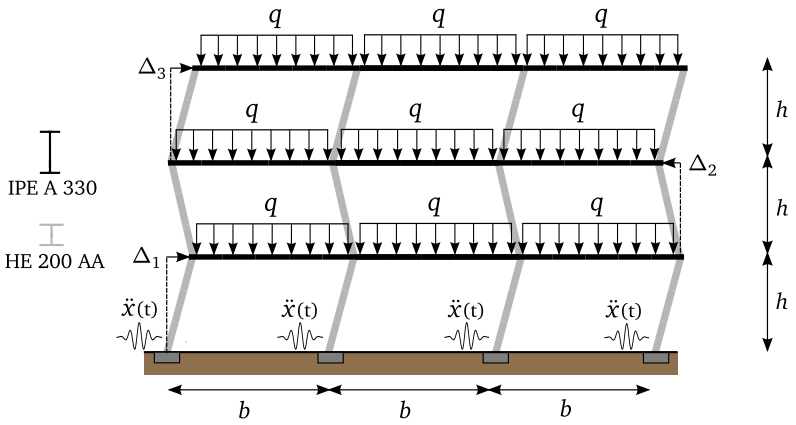


Figure 6: Three-story steel frame under ground motion excitation $\ddot{x}(t)$ and additional loading q. (Figure adapted from Zhu et al. (2023))

surrogate models, such as the recently introduced mNARX Schär et al. (2024), could in principle address this type of behavior, but it remains outside the scope of this work.

In total, we selected a rich set of 1,502 acceleration recordings of real earthquakes sampled at 200 Hz, covering near and far field, as well as low and high magnitude earthquakes. The ground motion velocity and displacement were obtained by direct numerical integration of the acceleration data. The simulations were performed using the open-source software OpenSees Pacific Earthquake Engineering Research Center (ease), and the resulting dataset was then randomly split into an experimental design comprising 100 realizations and an out-of-sample test set of 1,402 realizations for validation and performance assessment.

Table 3: Three-story steel frame – Frame parameters

Parameter	Unit	Value
Height h	m	3
Width b	m	5
Young's modulus E	MPa	$2.05 \cdot 10^5$
Yield stress f_y	MPa	235
Live load q	kN/m	20

4.2.1 Model configurations

The configurations of the three final surrogates are shown in Table 4. These configurations were obtained by following a basis adaptive scheme as described in Section 4.1.1 using the hybrid LARS algorithm from Section 3.3. We tested polynomial degrees (d) from 1 to 3, interaction orders (r) from 1 to 3, and q-norms of 0.7, 0.85, and 1.0. We evaluated the forecast performance

during LARS (Eq. (39)) at every 10-th iteration and stopped at a maximum of 500 LARS iterations. Further, we only used a subset of 100,000 samples of the full design matrix to reduce the computational cost during the model fitting process. The explained variances ν_i and memories T_i were chosen identically for all three surrogates. They were also chosen homogeneously between the exogenous inputs and the autoregressive input, so we will subsequently omit the subscript. It is worth noting that the number of principal components required to achieve $\nu = 0.95$ is considerably higher for the exogenous acceleration input $\ddot{x}$ than for the velocity $\dot{x}$ and displacement x. This is expected, since acceleration data carries higher frequencies than the corresponding velocities and displacements. The best configuration for all the surrogates has a polynomial degree of three but only for $\Delta_2(t)$ interaction terms were included. Remarkably, the final surrogate for $\Delta_3(t)$ is significantly sparser (only 19 non-zero coefficients) compared to the other two surrogates.

Table 4: Three-story steel frame – Configurations of the three automatically selected $\mathcal{F}$-NARX models

Quantity of interest	$\Delta_1(t)$	$\Delta_2(t)$	$\Delta_3(t)$
Memory (T)	1.5 s	1.5 s	1.5 s
Explained variances (ν)	0.95	0.95	0.95
# principal components ($n_{c,\ddot{x}}$,$n_{c,\dot{x}}$,$n_{c,x}$,$n_{c,\Delta}$)	40/4/2/4	40/4/2/3	40/4/2/6
Maximum polynomial degree (d)	3	3	3
Maximum interaction order (r)	1	2	1
Q-norm (q)	1.0	0.7	1.0
# non-zero coefficients / # coefficients	146/150	127/1,323	19/156

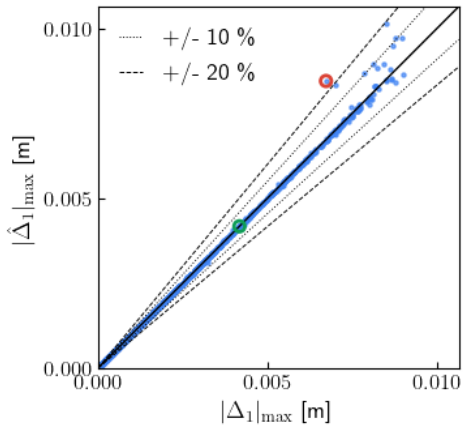
4.2.2 Results

The results of the $\mathcal{F}$-NARX surrogates on the three-story steel frame case study are shown in Figures 7-9. Figure 7a compares the predictions of the $\mathcal{F}$-NARX model on the out-of-sample set of simulations for the 1$^{\text{st}}$ floor interstory drift, in terms of absolute peak drift, a quantity traditionally extremely difficult to approximate due to its highly non-linear and phase-sensitive nature. It can be seen that the absolute peak interstory drifts of the predicted traces align well with the reference ones. For most simulations, the discrepancy is within the 10 % error bounds, with only a handful of outliers with up to 30 % error in the far right tail. Figure 7b shows the corresponding survival plot with respect to a critical maximum drift $|\Delta|^{\text{crit}}_{\text{max}}$, calculated from the true and predicted drifts. The predicted curve matches the reference up to an exceedance probability of about $3 \cdot 10^{-3}$, with a mild deviation for lower probabilities. For reference, the same curve computed from the 100 training traces is provided, and as expected it clearly deviates from the reference already for relatively high exceedance probabilities because of the limited dataset. For visualization purposes, we also show two traces of two example simulations in Figure 7c,

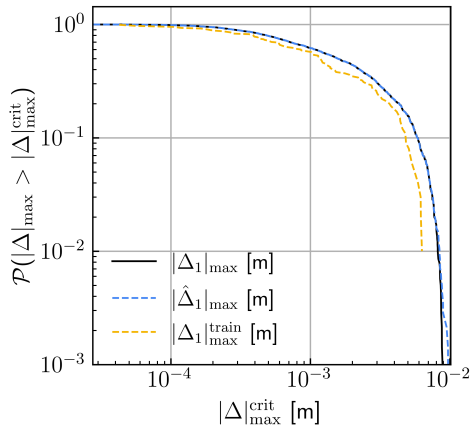
in which the left and right traces have a low and high prediction error in $|\Delta|_{\max}$, respectively. The predicted trace with the low error aligns almost perfectly with the reference over the full simulation duration, while the other is generally stable over the full signal duration, but shows spurious higher frequency oscillations in the range of the extreme drift values.

Figures 8a-c show the same results for the 2^{nd} floor interstory drift. The results are mostly comparable to the ones for the 1^{st} floor, albeit with an overall lower accuracy, especially in the high tail of the distribution. The discrepancy in $|\Delta|_{\max}$ is significantly higher, which is reflected in the higher dispersion of the points in Figures 8a and the earlier divergence between the true and predicted survival curves in Figures 8b. The worse performance on the 2^{nd} floor interstory drift may be due to more complex dynamics, which are also reflected in the higher polynomial degree and interaction terms listed in Table 4.

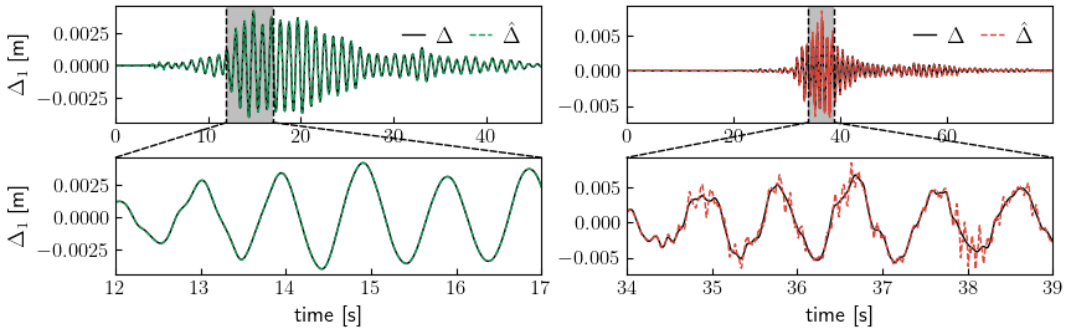
Finally, we show the results for the 3^{rd} floor interstory drift in Figures 9a-c, which show an overall better agreement to the reference with respect to the previous two cases. The fact that the surrogate predicting the 3^{rd} floor has only 19 non-zero model coefficients (see Table 4) indicates that the dynamics of this floor are indeed somewhat less complex to predict than the others.



(a) Predicted vs. true peak absolute 1st floor interstory drift

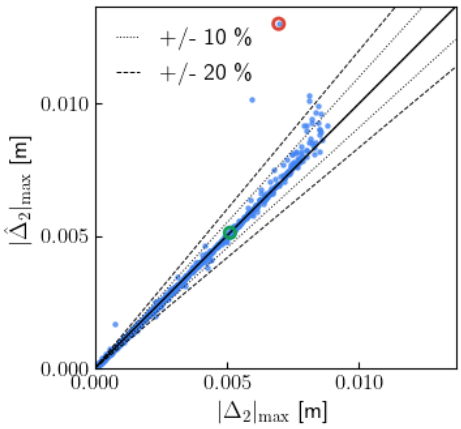


(b) Survival curve for the peak absolute 1st floor interstory drift

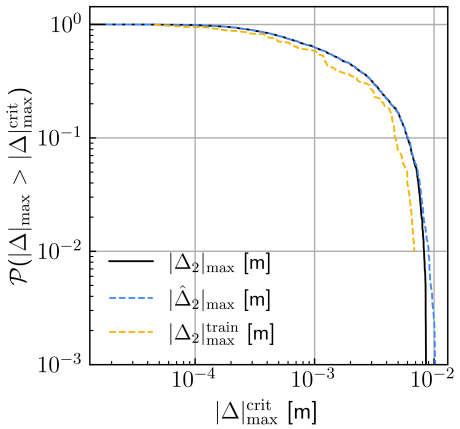


(c) Example traces of the true (solid black lines) vs. predicted (dashed colored lines) 1st floor interstory drift

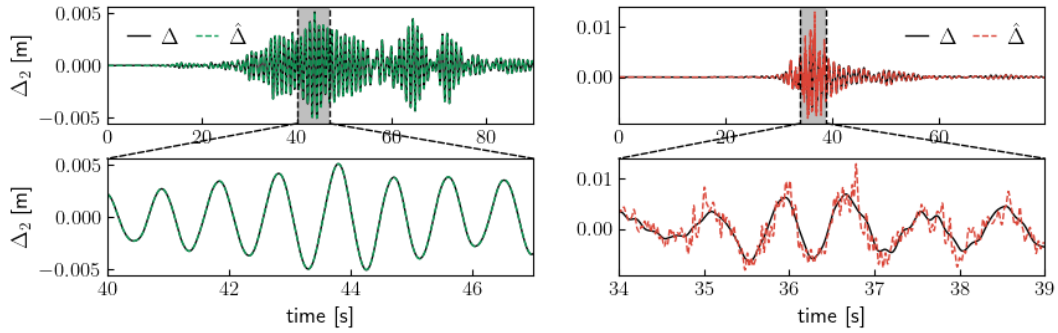
Figure 7: Three-story steel frame results. (Top left) Scatter plot of the true vs. predicted peak absolute interstory drift of the 1st floor. The solid black line indicates a perfect prediction whereas the remaining lines show the $10-20$ % error bounds. (Top right) Survival plot of the true and predicted peak interstory drifts. It shows the probability of $|\Delta|_{\max}$ exceeding a given threshold value $|\Delta|^{\text{crit}}_{\max}$. For reference the survival plot of the training dataset is shown in yellow. (Bottom) Example traces showing a good (green) and one of the worst (red) prediction. The green curve corresponds to a small relative error in the predicted absolute peak interstory drift. Analogously, the red curve has a high discrepancy in $|\Delta|_{\max}$. Note that these traces correspond to the green and red circles in the top left panel.



(a) Predicted vs. true peak absolute 2nd floor interstory drift

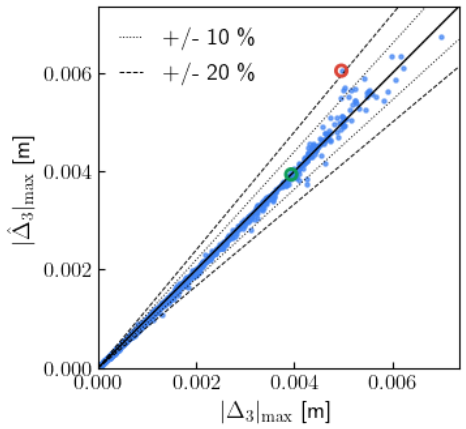


(b) Survival curve for the peak absolute 2nd floor interstory drift

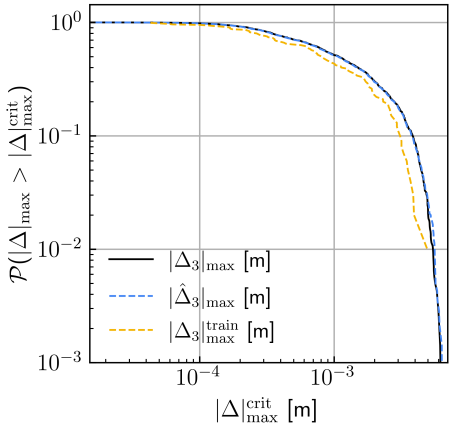


(c) Example traces of the true (solid black lines) vs. predicted (dashed colored lines) 2nd floor interstory drift

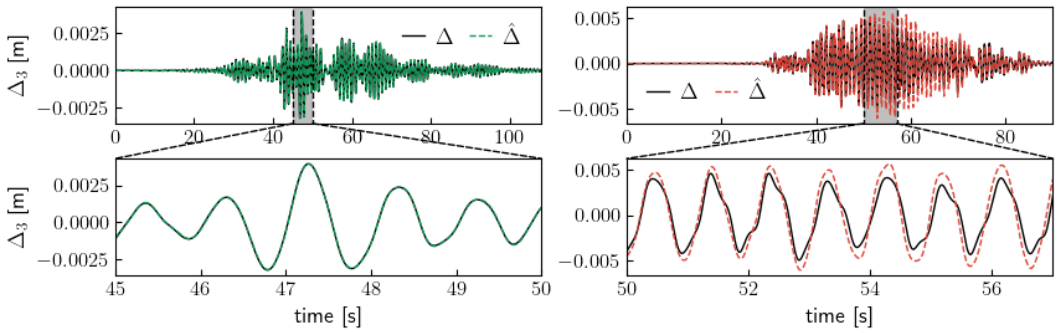
Figure 8: Three-story steel frame results. (Top left) Scatter plot of the true vs. predicted peak absolute interstory drift of the 2nd floor. (Top right) Survival plot of the true and predicted peak interstory drifts. (Bottom) Example traces showing one of the best (green) and one of the worst (red) predictions.



(a) Predicted vs. true peak absolute 3rd floor interstory drift



(b) Survival curve for the peak absolute 3rd floor interstory drift



(c) Example traces of the true (solid black lines) vs. predicted (dashed colored lines) 3rd floor interstory drift

Figure 9: Three-story steel frame results. (Top left) Scatter plot of the true vs. predicted peak absolute interstory drift of the 3rd floor. (Top right) Survival plot of the true and predicted peak interstory drifts. (Bottom) Example traces showing one of the best (green) and one of the worst (red) predictions.

5 Discussion and Conclusion

In this paper, we present a novel approach to nonlinear autoregressive with exogenous inputs (NARX) modeling, called $\mathcal{F}$-NARX for *functional* NARX. By following a continuous functional view point, rather than the well-established discrete-time approach, $\mathcal{F}$-NARX overcomes several limitations of traditional NARX models, making it particularly well-suited for surrogate modeling of dynamical physical systems, such as engineering structures, also in the presence of multiple exogenous inputs. $\mathcal{F}$-NARX addresses the sensitivity of classical NARX models to time discretization, their tendency to over-rely on recent autoregressive inputs, and their inefficiency in handling systems with long memory. These challenges are mitigated by modeling the system dynamics in a transformed space that captures key physical features from both the exogenous inputs and the autoregressive input (focus on functional features) rather than relying on original discrete time steps (focus on time discretization).

To validate the methodology, we chose a combination of feature extraction using principal component analysis and sparse polynomial NARX modeling based on hybrid LARS, and applied it to two different case studies intended to showcase both the robustness of the method w. r. t. the limitations of classical NARX, and its performance in the presence of a complex computational model.

In the first case study, we examined the behavior of $\mathcal{F}$-NARX applied to an eight-story building subjected to wind loading. The goal of this case study was to investigate the behavior of $\mathcal{F}$-NARX with respect to its configuration parameters. We demonstrated that the model is straightforward to parameterize using interpretable configuration parameters, and at the same time that it is largely unaffected by changes in the sampling rate of the simulations, enabling it to handle highly oversampled signals effectively.

In a challenging second case study, involving a nonlinear finite element model of a three-story steel frame under seismic loading, we demonstrated $\mathcal{F}$-NARX's capability to accurately model more complex dynamical systems. The $\mathcal{F}$-NARX models accurately predicted the interstory drifts of the building over extended periods, despite being trained on a small experimental design of only 100 simulations. This forecast stability is largely due to the modeling of the system in the transformed feature space, which reduces the over-reliance on individual recent past output time steps that are highly correlated with the current output value.

The $\mathcal{F}$-NARX methodology, when deployed using PCA, polynomial basis functions with a basis adaptive scheme, and a sparse solver, proved to be a powerful surrogate for modeling complex dynamical systems typical of engineering simulation. Moreover, this implementation maintains low computational costs during model forecasting, making it practical for surrogate modeling. For the more complex second case study, the cost of the entire model fitting, including basis adaptivity, was in the order of 6 hours on a standard notebook. This time strongly depends on the size of the experimental design, as most of the time is spent in the modified LARS algorithm to evaluate the forecast performance on the experimental design (see Section 3.3). The time

taken to predict the 1,402 validation traces was about 6 minutes, which is approximately two orders of magnitude faster than the $\sim$ 8 hours required to run the corresponding OpenSees simulations.

Future research can explore the integration of other feature extraction algorithms and their synergy with different basis functions. Additionally, leveraging prior system knowledge to construct features relevant to the system response is a promising avenue. As shown by Schär et al. (2024), incorporating expert knowledge into the modeling process can significantly improve the emulation of complex dynamical systems.

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.

Acknowledgments

This project is part of the *HIghly advanced Probabilistic design and Enhanced Reliability methods for high-value, cost-efficient offshore WIND* (HIPERWIND) project and has received funding from the European Union's Horizon 2020 Research and Innovation Programme under Grant Agreement No. 101006689.

The authors express their sincere appreciation to Jungho Kim, Sang-ri Yi and Junho Song for their valuable contribution to the first case study by providing the corresponding numerical codes.

References

Acuña, G., C. Ramirez, and M. Curilem (2012). Comparing NARX and NARMAX models using ANN and SVM for cash demand forecasting for ATM. In *The 2012 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–6.

Aguirre, L. and S. Billings (1993). Relationship between the structure and performance of identified nonlinear polynomial models. Research report.

Aguirre, L. A. (1994). Some remarks on structure selection for nonlinear models. *International Journal of Bifurcation and Chaos 4*(6), 1707–1714.

Aguirre, L. A. and S. A. Billings (1995). Improved structure selection for nonlinear models based on term clustering. *International Journal of Control 62*(3), 569–587.

Aguirre, L. A. and C. Jácome (1998). Cluster analysis of NARMAX models for signal-dependent systems. *IEE Proceedings - Control Theory and Applications 145*, 409–414(5).

Awtoniuk, M., M. Daniun, K. Sałat, and R. Sałat (2019). Impact of feature selection on system identification by means of NARX-SVM. *MATEC Web of Conferences 252*, 03012.

Ayala, J., H. Wei, and S. Billings (2017). A novel logistic-NARX model as a classifier for dynamic binary classification. *Neural Computing and Applications*.

Badeau, R., G. Richard, and B. David (2004). Sliding window adaptive SVD algorithms. *IEEE Transactions on Signal Processing 52*(1), 1–10.

Bastiaans, M. (1985). On the sliding-window representation in digital signal processing. *IEEE Transactions on Acoustics, Speech, and Signal Processing 33*(4), 868–873.

Bengio, Y., O. Delalleau, N. Le Roux, J.-F. Paiement, P. Vincent, and M. Ouimet (2006). *Spectral Dimensionality Reduction*, pp. 519–550. Springer Berlin Heidelberg.

Bhattacharyya, B., E. Jacquelin, and D. Brizard (2020). A Kriging–NARX model for uncertainty quantification of nonlinear stochastic dynamical systems in time domain. *Journal of Engineering Mechanics 146*(7), 04020070.

Billings, S. A. (2013). *Nonlinear system identification: NARMAX methods in the time, frequency, and spatio-temporal domains*. Chichester, West Sussex, United Kingdom: John Wiley & Sons, Inc.

Billings, S. A. and L. A. Aguirre (1995). Effects of the sampling time on the dynamics and identification of nonlinear models. *International Journal of Bifurcation and Chaos 5*(6), 1541–1556.

Blatman, G. and B. Sudret (2010). An adaptive algorithm to build up sparse polynomial chaos expansions for stochastic finite element analysis. *Probabilistic Engineering Mechanics 25*(2), 183–197.

Blatman, G. and B. Sudret (2011). Adaptive sparse polynomial chaos expansion based on least angle regression. *J. Comput. Phys. 230*(6), 2345–2367.

Bracewell, R. N. (1989). The Fourier transform. *Scientific American 260*(6), 86–95.

Chen, S., X. X. Wang, and C. J. Harris (2008). NARX-based nonlinear system identification using orthogonal least squares basis hunting. *IEEE Transactions on Control Systems Technology 16*(1), 78–84.

Chen, Z. H. and Y. Q. Ni (2011). On-board identification and control performance verification of an MR damper incorporated with structure. *Journal of Intelligent Material Systems and Structures 22*(14), 1551–1565.

Cheng, Y., L. Wang, M. Yu, and J. Hu (2011). An efficient identification scheme for a nonlinear polynomial NARX model. *Artificial Life and Robotics 16*(1), 70–73.

Chiras, N., C. Evans, and D. Rees (2001). Nonlinear gas turbine modeling using NARMAX structures. *IEEE Transactions on Instrumentation and Measurement 50*(4), 893–898.

Coca, D. and S. A. Billings (2001). Non-linear system identification using wavelet multiresolution models. *International Journal of Control 74*(18), 1718–1736.

Dassanayake, S., A. Mousa, G. J. Fowmes, S. Susilawati, and K. Zamara (2023). Forecasting the moisture dynamics of a landfill capping system comprising different geosynthetics: A NARX neural network approach. *Geotextiles and Geomembranes 51*(1), 282–292.

Deshmukh, A. P. and J. T. Allison (2017). Design of dynamic systems using surrogate models of derivative functions. *Journal of Mechanical Design 139*(10), 101402.

Efron, B., T. Hastie, I. Johnstone, and R. Tibshirani (2004). Least angle regression. *The Annals of Statistics 32*(2), 407 – 499.

Eldar, Y. and G. Kutyniok (2012). *Compressed Sensing: Theory and Applications*. Compressed Sensing: Theory and Applications. Cambridge University Press.

F. Bianchi, A. Falsone, M. P. and L. Piroddi (2017). A randomised approach for NARX model identification based on a multivariate Bernoulli distribution. *International Journal of Systems Science 48*(6), 1203–1216.

Falsone, A., L. Piroddi, and M. Prandini (2014). A novel randomized approach to nonlinear system identification. In *53rd IEEE Conference on Decision and Control*, pp. 6516–6521.

Farina, M. and L. Piroddi (2009). Simulation error minimization–based identification of polynomial input–output recursive models. *IFAC Proceedings Volumes 42*(10), 1393–1398.

Farina, M. and L. Piroddi (2010). An iterative algorithm for simulation error based identification of polynomial input–output models using multi-step prediction. *International Journal of Control 83*(7), 1442–1456.

Gao, Y., S. Liu, F. Li, and Z. Liu (2016). Fault detection and diagnosis method for cooling dehumidifier based on LS-SVM NARX model. *International Journal of Refrigeration 61*, 69–81.

Garg, S., H. Gupta, and S. Chakraborty (2022). Assessment of DeepONet for reliability analysis of stochastic nonlinear dynamical systems. arXiv:2201.13145.

Hu, Z., J. Fang, R. Zheng, M. Li, B. Gao, and L. Zhang (2024). Efficient model predictive control of boiler coal combustion based on NARX neutral network. *Journal of Process Control 134*, 103158.

Jolliffe, I. T. (2002). *Principal Component Analysis*. Springer-Verlag.

Kim, J., S. Yi, and J. Song (2023). Estimation of first-passage probability under stochastic wind excitations by active-learning-based heteroscedastic Gaussian process. *Structural Safety 100*, 102268.

Kim, Y. (2015). Prediction of the dynamic response of a slender marine structure under an irregular ocean wave using the NARX-based quadratic Volterra series. *Applied Ocean*

Research 49, 42–56.

Kocijan, J. (2012). Plenary lecture 1: Dynamic GP models: An overview and recent developments. In *Proceedings of the 6th International Conference on Applied Mathematics, Simulation, Modelling*, ASM'12, Stevens Point, Wisconsin, USA, pp. 12. World Scientific and Engineering Academy and Society (WSEAS).

Koziel, S., L. Leifsson, and X.-S. Yang (Eds.) (2014). *Solving Computationally Expensive Engineering Problems: Methods and Applications*, Volume 97. Springer International Publishing.

Lacerda Junior, W. R., S. A. M. Martins, and E. G. Nepomuceno (2021). Meta-model structure selection: Building polynomial narx model for regression and classification. arXiv:2109.09917.

Langeron, Y., K. T. Huynh, and A. Grall (2021). A root location-based framework for degradation modeling of dynamic systems with predictive maintenance perspective. *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability 235*(2), 253–267.

Lataniotis, C., S. Marelli, and B. Sudret (2020). Extending classical surrogate modeling to high dimensions through supervised dimensionality reduction: A data-driven approach. *International Journal for Uncertainty Quantification 10*(1), 55–82.

Levin, A. and K. Narendra (1996). Control of nonlinear dynamical systems using neural networks. II. Observability, identification, and control. *IEEE Transactions on Neural Networks 7*(1), 30–42.

Li, B., W.-C. Chuang, and S. M. J. Spence (2021). Response estimation of multi-degree-of-freedom nonlinear stochastic structural systems through metamodeling. *Journal of Engineering Mechanics 147*(11), 04021082.

Li, D., J. Zhou, and Y. Liu (2021). Recurrent-neural-network-based unscented Kalman filter for estimating and compensating the random drift of MEMS gyroscopes in real time. *Mechanical Systems and Signal Processing 147*, 107057.

Loève, M. (1955). *Probability Theory*. University series in higher mathematics. Springer-Verlag.

Mai, C.-V., M. D. Spiridonakos, E. Chatzi, and B. Sudret (2016). Surrogate modeling for stochastic dynamical systems by combining nonlinear autoregressive with exogeneous input models and polynomial chaos expansions. *Int. J. Uncertainty Quantification 6*(4), 313–339.

Mattson, S. G. and S. M. Pandit (2006). Statistical moments of autoregressive model residuals for damage localisation. *Mechanical Systems and Signal Processing 20*(3), 627–645.

Murray-Smith, R., T. A. Johansen, and R. Shorten (1999). On transient dynamics, off-equilibrium behaviour and identification in blended multiple model structures. In *1999 European Control Conference*, pp. 3569–3574. IEEE.

Olkkonen, J. T. (2011). *Discrete Wavelet Transforms - Theory and Applications*. IntechOpen.

Pacific Earthquake Engineering Research Center (Year of Software Release). *OpenSees: Open*

System for Earthquake Engineering Simulation. Berkeley, CA: University of California, Berkeley.

Pearson, K. (1901). LIII. *On lines and planes of closest fit to systems of points in space. The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science 2*(11), 559–572.

Piroddi, L. (2008). Simulation error minimisation methods for NARX model identification. *International Journal of Modelling, Identification and Control 3*, 392–403.

Piroddi, L., V. Seghezza, and M. Bonin (2010). NARX model selection based on simulation error minimisation and LASSO. *IET Control Theory & Applications 4*(7), 1157–1168.

Piroddi, L. and W. Spinelli (2003). An identification algorithm for polynomial NARX models based on simulation error minimization. *International Journal of Control 76*(17), 1767–1781.

Power, M., B. Chiou, N. Abrahamson, Y. Bozorgnia, T. Shantz, and C. Roblee (2008). An overview of the NGA project. *Earthquake Spectra 24*(1), 3–21.

Pulecchi, T. and L. Piroddi (2007). A cluster selection approach to polynomial NARX identification. In *2007 American Control Conference*, pp. 852–857.

Ramin, V. P., H. G. Amin, and N. Alireza (2023). An investigation of the performance of the ANN method for predicting the base shear and overturning moment time-series datasets of an offshore jacket structure. *International Journal of Sustainable Construction Engineering and Technology 14*(4), 79–93.

Ranković, V., N. Grujović, D. Divac, and N. Milivojević (2014). Development of support vector regression identification model for prediction of dam structural behaviour. *Structural Safety 48*, 33–39.

Rumelhart, D. E. and J. L. McClelland (1986). *Parallel distributed processing: explorations in the microstructure of cognition*. Computational models of cognition and perception. MIT Press.

Samsuri, N. A., S. A. Raman, and T. M. Y. S. Tuan Ya (2023). Evaluation of NARX network performance on the maintenance application of rotating machines. In F. Ahmad, H. H. Al-Kayiem, and W. P. King Soon (Eds.), *ICPER 2020*, Singapore, pp. 593–609. Springer Nature Singapore.

Schlechtingen, M. and I. Ferreira Santos (2011). Comparative analysis of neural network and regression based condition monitoring approaches for wind turbine fault detection. *Mechanical Systems and Signal Processing 25*(5), 1849–1875.

Schär, S., S. Marelli, and B. Sudret (2024). Emulating the dynamics of complex systems using autoregressive models on manifolds (mNARX). *Mechanical Systems and Signal Processing 208*, 110956.

Siegelmann, H., B. Horne, and C. Giles (1997). Computational capabilities of recurrent NARX neural networks. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) 27*(2), 208–215.

Song, H., X. Shan, L. Zhang, G. Wang, and J. Fan (2022). Research on identification and active vibration control of cantilever structure based on NARX neural network. *Mechanical Systems and Signal Processing 171*, 108872.

Spinelli, W., L. Piroddi, and M. Lovera (2006). A two-stage algorithm for structure identification of polynomial NARX models. In *2006 American Control Conference*, pp. 2387–2392.

Spiridonakos, M. D. and E. N. Chatzi (2015). Metamodeling of nonlinear structural systems with parametric uncertainty subject to stochastic dynamic excitation. *Earthquakes and Structures 8*(4), 915–934.

Strang, G. and T. Nguyen (1996). *Wavelets and Filter Banks*. Philadelphia, PA: Wellesley-Cambridge Press.

Sundararajan, D. (2001). *The Discrete Fourier Transform: Theory, Algorithms and Applications*. World Scientific.

Tenenbaum, J. B., V. de Silva, and J. C. Langford (2000). A global geometric framework for nonlinear dimensionality reduction. *Science 290*(5500), 2319–2323.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological) 58*(1), 267–288.

Verleysen, M. and D. François (2005). The curse of dimensionality in data mining and time series prediction. In *Proceedings of the 8th International Conference on Artificial Neural Networks: Computational Intelligence and Bioinspired Systems*, IWANN'05, Berlin, Heidelberg, pp. 758–770. Springer-Verlag.

Wei, H. L. and S. A. Billings (2008). Model structure selection using an integrated forward orthogonal search algorithm assisted by squared correlation and mutual information. *International Journal of Modelling, Identification and Control 3*(4), 341.

Worden, K. and R. J. Barthorpe (2012). Identification of hysteretic systems using narx models, part I: Evolutionary identification. In *Topics in Model Validation and Uncertainty Quantification, Volume 4*, pp. 49–56. Springer New York.

Worden, K., W. Becker, T. Rogers, and E. Cross (2018). On the confidence bounds of Gaussian process NARX models and their higher-order frequency response functions. *Mechanical Systems and Signal Processing 104*, 188–223.

Wunsch, A., T. Liesch, and S. Broda (2018). Forecasting groundwater levels using nonlinear autoregressive networks with exogenous input (NARX). *Journal of Hydrology 567*, 743–758.

Yetkin, M., M.-S. Kim, and Y. Kim (2017). Mooring line top-tension prediction using NARX. International Ocean and Polar Engineering Conference, pp. ISOPE–I–17–375.

Yu, C., Y.-P. Zhu, H. Luo, Z. Luo, and L. Li (2023). Design assessments of complex systems based on design oriented modelling and uncertainty analysis. *Mechanical Systems and Signal Processing 188*, 109988.

Yuzhu Guo, L.Z. Guo, S. B. and H.-L. Wei (2015). An iterative orthogonal forward regression algorithm. *International Journal of Systems Science 46*(5), 776–789.

Zhang, J., Z. Yin, and R. Wang (2017). Nonlinear dynamic classification of momentary mental workload using physiological features and NARX-model-based least-squares support vector machines. *IEEE Transactions on Human-Machine Systems 47*(4), 536–549.

Zhang, L., S. Draycott, and P. Stansby (2024). System identification and generalisation of elastic mooring line forces on a multi-float wave energy converter platform in steep irregular waves. *Mechanical Systems and Signal Processing 214*, 111259.

Zhang, Y., Y. Dong, and M. Beer (2024). rLSTM-AE for dimension reduction and its application to active learning-based dynamic reliability analysis. *Mechanical Systems and Signal Processing 215*, 111426.

Zhou, J. and J. Li (2023). An efficient time-variant reliability analysis strategy embedding the NARX neural network of response characteristics prediction into probability density evolution method. *Mechanical Systems and Signal Processing 200*, 110516.

Zhu, X., M. Broccardo, and B. Sudret (2023). Seismic fragility analysis using stochastic polynomial chaos expansions. *Probabilistic Engineering Mechanics 72*(103413).