Reconstruction of governing equation for nonlinear dynamical systems based on Universal Differential Equation

José G. Cóndor López[1,3,*], Michael Leupolz[1,2], Sven Herold[3]

[1]Mercedes-Benz Group AG; michael.leupolz@mercedes-benz.com
[2]Karlsruhe Institute of Technology, Institute of Vehicle System Technology
[3]Fraunhofer Institute for Structural Durability and System Reliability LBF; sven.herold@lbf.fraunhofer.de
[*]Correspondence: jose_gabriel.condor_lopez@mercedes-benz.com

Abstract: In the sense of black-box approaches, data-driven models can be used for the mathematical description of complex dependencies of (multi-)physical processes. A specific or prior physical knowledge inside the model is not required and is compensated by cost-intensive data amounts. Due to the restrictive accessibility of data in some engineering fields, black-box models are limited regarding their applicability. The incorporation of physical knowledge into Machine Learning methods counteracts data limitations and leads to data efficient modelling approaches. The Universal Differential Equation (UDE) approach seeks for data reduction by combining physical based models and Machine Learning. This enables semi-automated and in some special cases fully-automated modelling. The resulting models and their evaluation are valuable for gaining knowledge during development, production and operating phase. In this paper, UDE is applied to reconstruct the governing equation of a nonlinear dynamical system represented by a forced duffing oscillator and compared with black-box approaches afterwards. In contrast to the approximation of the entire governing equation by an Universal Approximator, UDE aims to approximate only unknown terms inside the differential equation using Universal Approxiamtors such as Neural Networks. Based on sparse regression methods these unknown terms are reconstructable in a targeted manner. The methodology is applied and validated on a nonlinear dynamical system considering robustness and sensitivity aspects against uncertainty-prone training data (e.g. measurement data) and different data-driven modelling approaches are compared regarding their forecast capabilities. Afterwards, potentials for further fields of application in automotive development are shown.

Keywords: Scientific Machine Learning, Universal Differential Equation, forecasting, equation reconstruction, nonlinear dynamics, sparse regression

1 Introduction

In many science and engineering applications numerical simulations of physical based models described by ordinary or partial differential equation are powerful tools regarding design space exploration and prediction capability of dynamical system behaviour. Investigation such as sensitivity analysis, identifying optimal model parameters or predictions beyond existing measurement data are enabled by these kind of physical models [1]. However, in some engineering fields like automotive or aerospace sector the computation of these models is characterized by complex and resource intensive numerical simulations [2, 3]. Due to the rapidly increasing interest on Machine Learning (ML) in the last years and their impressive results in application fields where large amount of data are collected [3], different ML based models are applied to approximate the system behaviour in a time consuming manner. Whereas in [4] Neural Networks (NN) are used to predict noise transfer functions depending on excitation and geometry parameters, Gaussian Process Regression is proposed to approximate vehicle crash finite-element simulations [2]. The presented metamodel in [5] is applied to learn the dependency between geometrical kinematics manipulation of a chassis component and their resulting characteristic frequency response function. Such a metamodel is able to identify optimal geometry configurations regarding the transfer behaviour. These data-driven models represent black-box models due to their ability of mapping inputs directly to outputs without any scientific knowledge inside the model. Furthermore, they are trained on a

set of numerical simulation results. In some applications these amount of data is not accessible or little to no data exist due to expensive experiments [1, 6]. Therefore, new approaches are needed to overcome these data limitations. By integrating ML approaches into physical based models the field of Scientific Machine Learning (SciML) aims for data amount reduction as well as acceleration during training and prediction which shows potential in different engineering applications [1, 3, 6]. Whereas in [7] ML concepts are combined with scientific structures resulting in a new kind of NNs known as Neural Ordinary Differential Equations (NODE) which represents an initial value problem, the proposed Physics-Informed Neural Network (PINN) in [8] incorporates prior physical knowledge in the form of partial differential equation into the loss function. The presented approach Universal Differential Equation (UDE) in [6] combines ML techniques selectively inside the prior known physical based model to consider and approximate unknown physical dependencies of the model. By means of sparse regression, these approximated unknown terms are reconstructable afterwards. This pointwise identification requires less state information in comparison to the presented Sparse Identification of Nonlinear Dynamics (SINDy) approach by [9] which generates proper results recovering ordinary differential equations (ODE) in [10, 11, 12].

This work aims to compare different data-driven modelling approaches in time domain regarding different depth of physical knowledge incorporation and their forecasting capabilities based on a nonlinear dynamical system. Afterwards, the potential of UDE regarding reconstruction of unknown terms is discussed.

2 Data-driven modelling approaches

In the following, we introduce different data-driven modelling approaches with the ability to model dynamical system behaviour in time domain and to forecast time series. The approaches differ regarding their depth of incorporating physical knowledge inside the model and cover fields of classical ML and SciML. The presented approaches are trained using the Python package PyTorch and the Julia package DiffEqFlux.

2.1 Nonlinear Autoregressive Neural Network

The main idea of the Nonlinear Autoregressive (NAR) is based on the extension of the linear autoregressive model proposed by [13]. Instead of using linear combinations the NAR approach integrates a nonlinear estimation function to approximate the time series by a previous sequence of time series elements k which is also known as time window [14]. Due to the capabilities of NN to approximate any function with a desired accuracy stated by the Universal Approximator (UA) theorem in [15], we introduce a NAR model with NN as nonlinear estimation function f_{NN} described by

$$\mathbf{x}(t) = f_{\mathrm{NN}}\left(\mathbf{x}_{t-1}, \mathbf{x}_{t-2}, \dots, \mathbf{x}_{t-k}, \mathbf{p}\right). \tag{1}$$

Here, $\mathbf{x}(t) \in \mathbb{R}^{s\times 1}$ represents the s states of the system at time t and $\mathbf{p}$ represents the weights and biases of the NN which is a feedforward net with $s \times k$ inputs and s outputs.

2.2 Long Short-Term Memory

Long Short-Term Memory (LSTM) is a special form of recurrent NNs, explicitly designed to predict time series with large time lag. To make this possible a so-called memory cell is embedded in the NN. The memory cell consists of a self-recurrent connection (SRC) and two gates known as input and output gates, positioned before and after the SRC. Those gates decide which data to store in the memory cell and whether to apply the included information on the next input data of the NN. As gates, sigmoid neural net layers are used. This form of LSTM was first introduced in. [16]

Over the years many variants of LSTMs were developed. An overview of different variations can be found in [17]. The LSTM used in this research includes an additional gate. The forget gate acts directly on the SRC and ensures that memory cell does not continuously grow but also actively resets [18]. A profound technical analysis is given in [19]. To enable the model to learn even more complex relations between input and output, multiple LSTM layers can be stacked, leading to one LSTM layer being the input to another LSTM layer. This procedure is e.g. investigated in [20].

2.3 Neural Ordinary Differential Equation

The proposed kind of deep NN models in [7] based on combining fields of ML with the numeric of differential equations solvers. Instead of approximate the nonlinear behaviour of the dynamical system directly without integrating any knowledge of the underlying system, a NN is introduced to learn the derivative of the system state and results in an initial value problem described by

$$\frac{d\mathbf{x}}{dt} = f_{\mathrm{NN}}\left(t, \mathbf{x}, \mathbf{p}\right) \quad \text{with} \quad \mathbf{x}\left(t=0\right) = \mathbf{x}_0, \tag{2}$$

where $\mathbf{x}_0$ denotes the initial condition. The analytical solution of the initial value problem is given by

$$\mathbf{x} = \mathbf{x}_0 + \int_{t_0}^{t} \left(f_{\mathrm{NN}}\left(t, \mathbf{x}, \mathbf{p}\right) \right) dt. \tag{3}$$

In the case where the analytical integration of $f_{\mathrm{NN}}\left(t, \mathbf{x}, \mathbf{p}\right)$ is unfeasible, numerical methods like forward Euler method are requiered to approximate the solution. The discretization is described by

$$\mathbf{x}_{t+1} = \mathbf{x}_t + h \cdot f_{\mathrm{NN}}\left(t, \mathbf{x}_t, \mathbf{p}\right), \tag{4}$$

where h corresponds to the step size and $\mathbf{x}_t$ as well as $\mathbf{x}_{t+1}$ represent the solution of the initial value problem at time t and $t+1$. The architecture of this resulting hybrid model built up by a NN and ODE solver is similar to a residual NN. This new kind of deep NN models called Neural ODE parameterize the derivative of the hidden state using NNs instead of specifying a discrete sequence of hidden layers [7]. Therefore, Neural ODE can be interpreted as an infinitely deep model [21]. Furthermore, any stated ODE solver can be used to solve the initial value problem.

2.4 Universial Differential Equation and sparse regression

The presented UDE approach in [6] corresponds to the emerging field of SciML and aims for integrating ML approaches into physical based models. The main idea of UDE is to consider unknown parameters or physical dependencies inside ODEs by incorporating UAs. Additionally, the missing equation fractions can be reconstructed based on the trained UAs afterwards. In comparison to NAR or LSTM, which approximate the dynamical behaviour without any specific physical information or PINN by inserting the complete ODE into the loss function [8], UDE merges both concepts and combines already physical knowledge with unknown term into a new kind of differential equation. In general, the UDE is formulated as follows:

$$\frac{d\mathbf{x}}{dt} = f\left(t, \mathbf{x}, \mathbf{u}\left(t, \mathbf{x}, \mathbf{p}\right)\right) \quad \text{with} \quad \mathbf{x}\left(t=0\right) = \mathbf{x}_0. \tag{5}$$

Here, f represents a function which describes the systems equation of motion and $\mathbf{u}$ denotes an UA, such as a NN, a Fourier expansion or other ML models. The selection of an appropriate UA depends on the dimensionality of the stated problem formulation [6]. This introduced initial value problem is solved analogously to the NODE approach, which can be therefore interpreted as a special form of UDE by treating the whole differential equation as unknown.

The reconstruction of unknown terms in **Equation** (5) based on the idea of the proposed methodology SINDy by [9]. Instead of reconstructing the entire differential equation of a dynamical system, we focus on

identifying unknown terms only which are described by UAs. The incorporation of the physical knowledge enables a more selectively equation reconstruction and the data collection of all states as well as their derivatives is not mandatory. The resulting regression problem is described by the system of equations

$$\mathbf{U} = \boldsymbol{\Theta}(\mathbf{X}) \cdot \boldsymbol{\Xi}, \tag{6}$$

where $\mathbf{U}$ denotes the prediction of the UA, $\mathbf{X}$ is a matrix representation of $\mathbf{x}$ and $\boldsymbol{\Xi}$ denotes the sparse vector of coefficients containing information about which nonlinearities are active. The library $\boldsymbol{\Theta}(\mathbf{X})$ consists of possible nonlinear functions [9] and can be build up by constant, polynomial or trigonometric terms like $\boldsymbol{\Theta} = \left[1, \mathbf{X}, \mathbf{X}^2, \mathbf{X}^3, \sin(\mathbf{X})\right]$. By inserting a L^1 regularization term to the regression problem in **Equation** (6), the sparsity of the coefficient vector $\boldsymbol{\Xi}$ is enhanced. The degree of sparsity is determined by the sparsification parameter λ thresholding all coefficients in $\boldsymbol{\Xi}$ below a certain cut off value [9]. The optimization regression problem results in:

$$\boldsymbol{\Xi} = \underset{\boldsymbol{\Xi}'}{\operatorname{argmin}} \left\| \boldsymbol{\Theta}(\mathbf{X})\, \boldsymbol{\Xi}' - \mathbf{U} \right\|_2 + \lambda \left\| \boldsymbol{\Xi}' \right\|_1 . \tag{7}$$

3 Problem definition

The study of the presented data-driven approaches are based on an external excited duffing oscillator system. The governing differential equation of this geometrically nonlinear dynamical system (NDS) is characterized by a nonlinear cubic stiffness term and is transformed into a first order ODE system described by

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= -\frac{1}{m}\left(c \cdot x_2 + k \cdot x_1 + \beta \cdot x_1^3 - A \cdot \sin(\omega \cdot t)\right), \end{aligned} \tag{8}$$

with the mass of the system m, the damping coefficient c, the stiffness coefficient k, the cubic stiffness coefficient β, the amplitude A and the frequency ω of the external applied harmonic force $F_{\text{ext}} = A \cdot \sin(\omega t)$. The states of the systems are represented by x_1 and x_2, respectively, $\dot{x}_1$ and $\dot{x}_2$ represent the time derivatives of the states. The mechanical model is defined as shown in **Figure** 1. The considered model parameters for the numerical simulation are selected to avoid periodic steady state behaviour during observed simulations time $T = [0, 10]$ and are summarized in **Table** 1. Further simulation parameters are settled up such that the system starts at rest position described by the initial conditions $x_1(t = 0) = 0$ and $x_2(t = 0) = 0$. For purposes of clarity, we introduce model parameters regardless of physical units.

The resulting ideal time series as well as the noise-affected series of NDS denoted by the displacement x_1 and the velocity x_2 is depicted in **Figure** 1. For the time integration we used $t_\Delta = 0.01$ as simulation

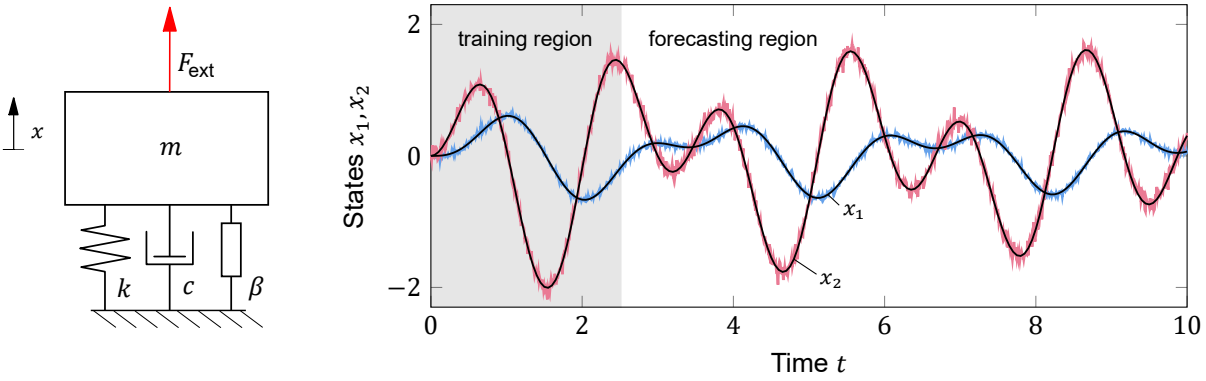

Figure 1: Description of the mechanical model of an external excited duffing oscillator and visualisation of resulting ideal ($\eta = 0$) and noisy ($\eta = 0.04$) time series of NDS. Division of the data into accessible (training region) and non-accessible data (forecasting region) for the data-driven approaches.

stepsize. The noise-manipulated time series is obtained by adding a normally distributed number characterized by zero mean value and unit standard deviation to each time step. The noise magnitude is adjustable by η. To obtain the training data for the presented data-driven approaches the time series is divided into two regions. The grey coloured area is introduced as training region and is accessible data during training phase. The time series in the forecasting region is utilized to evaluate the forecasting capabilities of the trained models afterwards. Therefore, this region represents non-accessible data. In some data-driven approaches, such as NN, testing data is required during training phase. These testing data will be selected from the training region as well.

Table 1: Model parameters of external excited geometrically nonlinear dynamical system

m	c	k	β	A	ω
1	0.1	4	0.5	3	4

In many engineering application fields such as automotive industry, physical model parameters like mass properties or stiffnesses of dynamical systems can be determined by measurements with a high accuracy. However, the identification of damping coefficients is characterized by uncertainty. For sake of simplicity, we assume a constant damping coefficient inspired by [11]. Therefore, we focus on the geometrically nonlinear term $\beta \cdot x_1^3$ as unknown physical dependency. To cover the unknown term inside the physical based model we augment them with an UA resulting in an UDE described by

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= -\frac{1}{m}\left(c \cdot x_2 + k \cdot x_1 - A \cdot \sin\left(\omega \cdot t\right)\right) + u\left(t, \mathbf{x}, \mathbf{p}\right). \end{aligned} \tag{9}$$

4 Results and comparison

The presented data-driven approaches in **Section** 2 are trained based on the noise-manipulated time series within the training region using a noise magnitude of $\eta = 0.04$ considering uncertainty. As UA, we used NN for each approach to model the dynamical behaviour of the system. The architecture as well as the activation functions of the implemented NN vary depending on the used approach and number of inputs. The evaluation of the trained models based on the mean squared error loss function $\mathcal{L}$ which is described by

$$\mathcal{L} = \frac{1}{n}\sum_j \mathcal{L}_j = \frac{1}{n}\sum_j \sum_i \left(x_{j,\text{train}}^i - x_{j,\text{pred}}^i\right)^2, \tag{10}$$

with the number of data points n, the training data $x_{j,\text{train}}^i$ and the prediction of the data-driven model $x_{j,\text{pred}}^i$. The index i denotes the i-th component whereas j denotes the j-th state of the respectively data. The selection of the NN architecture for each approach is motivated by identifying comparable data-driven models within a similar range of loss value moreover than seeking for the least loss value.

For the comparison and evaluation of the approaches, we depict the predicted time series of the models in **Figure** 2 divided into their system states. The corresponding loss values are listed in **Table** 2 classified regarding the considered region. The resulting loss values of the corresponding models inside the training region are limited by $\mathcal{L} < 0.01$ and is also clearly observable by comparing the time series trajectories in this region. Respectively, all trajectories are closely stacked and only minor deviations resulting from the LSTM are notable. Therefore, the selected data-driven model architectures are able to learn the dynamical behaviour of the system accurate enough considering noise-manipulated data and show non-overfitting properties.

In spite of similar loss values inside the training regions, the forecasting capabilities represented by the trajectories in $t \in (2.5, 10]$ show different deviations depending on the model. Whereas the UDE approach is capable to predict the physical behaviour with a loss value of $\mathcal{L}_{\text{UDE}} = 0.0066$, the forecasting

Table 2: Loss values of resulting data-driven approaches NAR, LSTM, NODE and UDE divided into training and forecasting region.

Loss value	NAR	LSTM	NODE	UDE
Training region	0.0051	0.0099	0.0052	0.0051
Forecasting region	2.9230	2.3974	3.4737	0.0066

of the models without physical incorporation NAR, LSTM and NODE are characterized by periodic and shifted time series trajectories. They are not predicting the dynamical behaviour in a right manner with comparatively large loss values $\mathcal{L}_{\text{LSTM}} \geq 2.3974$. While NAR and NODE are quite similar regarding periodicity of their trajectories, LSTM shows up less amplitude level and shifting properties, which leads to the second best result during the prediction although LSTM reaches the worst loss value during training. As a consequence, the loss value of the trained models is not an adequate indicator for the prediction performance of a model.

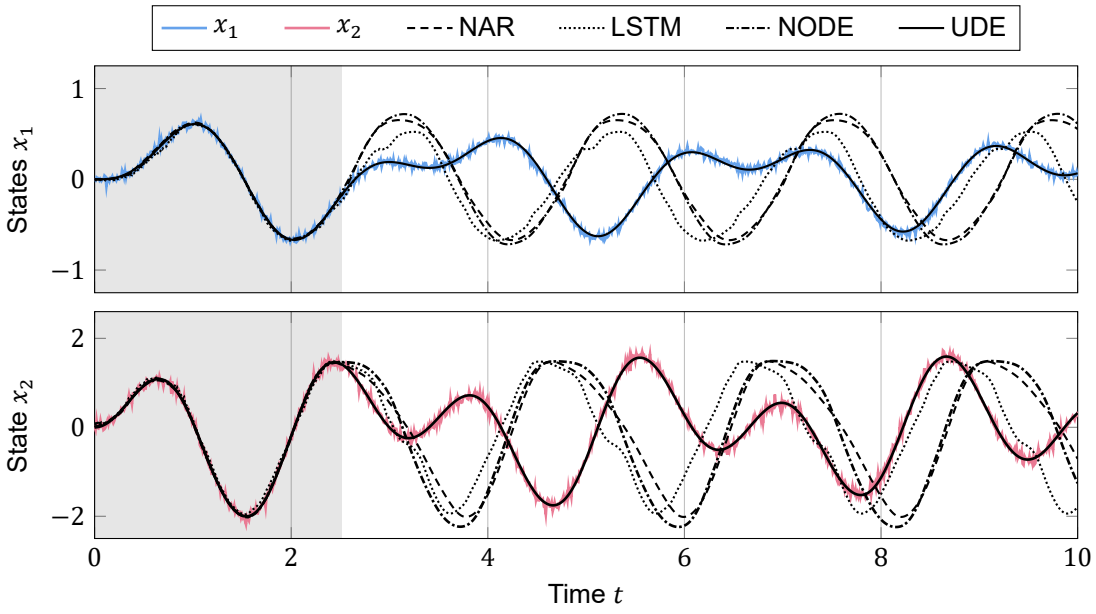

Figure 2: Comparison of the resulting data-driven appoaches based on noise-contaminated training data ($\eta = 0.04$) regarding training region ($t \in [0, 2.5]$) and forecasting capability ($t \in (2.5, 10]$).

The preserving periodicity of the non-physical incorporated models is substantiated by the time series during training, which tends to appear like a periodic curve. This erroneously pattern is learned by the data-driven approaches and mislead to undesired forecasting abilities. In that sense, the quality of the prediction depends strongly on the quantity of accessible data. It is assumable that, the more recorded amount of data during training is available, the better behaviour of dynamical system can be learned and predicted. Even though UDE is capable to model the missing physical dependency due to the incorporation of the already known physical information under same data conditions. Instead of learning the entire system, the NN inside the UDE has to learn a small fraction of the dynamical system and is able to forecast the time series in a proper way. Therefore, UDE demonstrates robust properties regarding limited amount of data as well as their noise-manipulation.

Due to the promising forecasting capabilities of the UDE approach, we use them for the selectively reconstruction of the missing term based on sparse regression presented in **Equation** 6. Therefore, we construct a library $\boldsymbol{\Theta}$ built up by polynomial functions up to order five as possible nonlinear function candidates. As sparsification parameter, we applied a range of permissible values $\lambda \in [0.1, 10]$ to find the pareto optimal solution. To provide better comparability, in **Figure** 3 the missing term $-\beta/m \cdot x_1^3$ is depicted

against the approximation by NN and the sparse reconstructed term for noise-manipulated ($\eta = 0.04$) and noiseless ($\eta = 0$) data sets. This missing term is usually non-accessible data during model reconstruction.

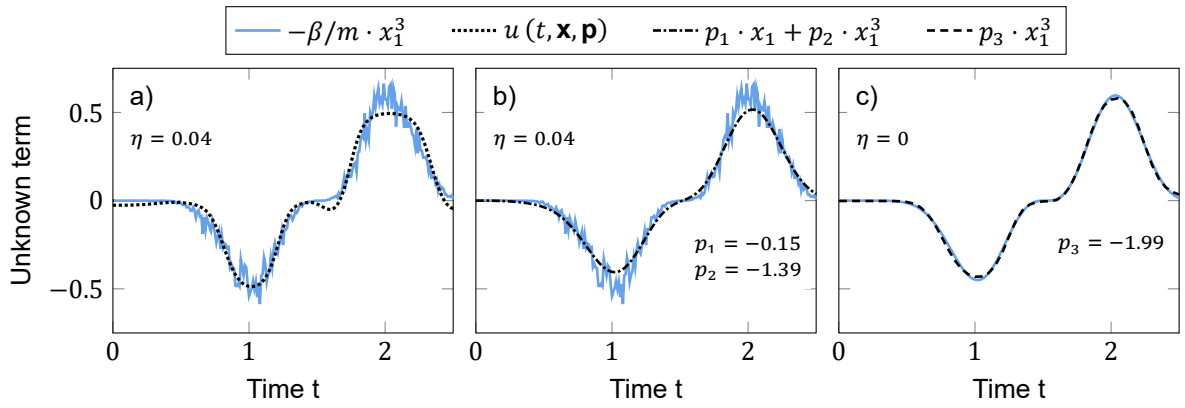

Figure 3: Comparison of noise-contaminated ($\eta = 0.04$) and ideal ($\eta = 0$) missing term with trained NN inside UDE (a), reconstructed equation (b) and ideal reconstructed equation (c) from sparse regression.

Even though UDE predicts the time series based on the noise-manipulated data accurate enough, the comparison of the missing term and the NN in **Figure** 3a shows up major deviations. The resulting reconstructed term of the sparse regression $u = p_1 \cdot x_1 + p_2 \cdot x_1^3$ is determined by a linear combination of polynomials where p_1 and p_2 are the active coefficients of the sparse vector. In **Figure** 3b there are still some notable deviations caused by the noisy data but the overall physical behaviour is represented in the right manner and leads to a slightly better performance regarding forecasting represented by the loss value of the reconstructed equation $\mathcal{L}_{\text{rec}} = 0.0064$ in comparison to that of purely NN inside UDE $\mathcal{L}_{\text{UDE}} = 0.0066$. In an ideally noiseless case, see **Figure** 3c, it is possible to fully reconstruct the missing term based on the trained NN. It should be mentioned that depending on the choice of function library, the reconstructed term might vary with respect to their sparsity. A restriction of possible function candidates in the sense of physical feasibility is therefore necessary to enable semi-automated or fully-automated modelling.

5 Conclusions and outlook

In this work, we present different data-driven approaches to model the dynamical behaviour of an external excited nonlinear oscillator in time domain regarding different depth of physical knowledge incorporation. By comparing them, we show that combining physical models with Machine Learning approaches results in more promising forecast capabilities than using ordinary forecasting algorithms even though all models show similar loss values during training phase. In addition, we demonstrate selectively term reconstruction of the missing term based on sparse regression. The usage of noise-contaminated data for the training of the approaches considers robustness aspects. This kind of approach enables the consideration of unknown physical dependencies by modelling them with surrogate models such as Neural Network and reconstruction of missing terms afterwards.

For future research, the Universal Differential Equation framework will be applied to more complex dynamical systems to explore the level of automation modelling in the sense of digital twin and will be confronted with challenges regarding appropriate Machine Learning architecture as well as function library during reconstruction. A specific application field in the automotive industry is the identification of unknown physical terms of electrified powertrain mount systems described by multi body models to improve the simulation e.g. the engine movement in different driving scenarios to cover package issues. Another application field is the extension of dynamical stiffness models of engine mounts in frequency domain by selectively integration of Universal Approximators in the sense of Universal Differential Equation.

References

[1] E. Y. Qian, “A Scientific Machine Learning Approach to Learning Reduced Models for Nonlinear Partial Differential Equations,” Ph.D. dissertation, Massachusetts Institute of Technology, Feb. 2021.

[2] J. Hay, J. Fehr, and L. Schories, Eds., *Crash Pulse Prediction for Scenario-based Vehicle Crash FE-Simulations*, ser. IRCOBI Conference 2020, no. IRC-20-22, Munich, Germany, 2020.

[3] N. Baker, F. Alexander, T. Bremer, A. Hagberg, Y. Kevrekidis, H. Najm, M. Parashar, A. Patra, J. Sethian, S. Wild, K. Willcox, and S. Lee, “Workshop Report on Basic Research Needs for Scientific Machine Learning: Core Technologies for Artificial Intelligence,” Feb. 2019.

[4] D. E. Tsokaktsidis, T. von Wysocki, F. Gauterin, and S. Marburg, “Artificial Neural Network predicts noise transfer as a function of excitation and geometry,” in *Proceedings of the 23rd International Congress on Acoustics*, Aachen, Germany, Sep 2019, pp. 4378–4382.

[5] T. von Wysocki, M. Leupolz, and F. Gauterin, “Metamodels Resulting from Two Different Geometry Morphing Approaches Are Suitable to Direct the Modification of Structure-Born Noise Transfer in the Digital Design Phase,” *Applied System Innovation*, vol. 3, no. 4, 2020.

[6] C. Rackauckas, Y. Ma, J. Martensen, C. Warner, K. Zubov, R. Supekar, D. Skinner, A. Ramadhan, and A. Edelman, “Universal Differential Equations for Scientific Machine Learning,” arXiv, 2021.

[7] R. T. Q. Chen, Y. Rubanova, J. Bettencourt, and D. Duvenaud, “Neural Ordinary Differential Equations,” arXiv, 2019.

[8] M. Raissi, P. Perdikaris, and G. Karniadakis, “Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations,” *Journal of Computational Physics*, vol. 378, pp. 686–707, 2019.

[9] S. L. Brunton, J. L. Proctor, and J. N. Kutz, “Discovering governing equations from data by sparse identification of nonlinear dynamical systems,” *Proceedings of the National Academy of Sciences*, vol. 113, no. 15, pp. 3932–3937, 2016.

[10] M. Stender, S. Oberst, and N. Hoffmann, “Recovery of differential equations from impulse response time series data for model identification and feature extraction,” *Vibration*, vol. 2, no. 1, pp. 25–46, 2019.

[11] M. Didonna, M. Stender, A. Papangelo, F. Fontanela, M. Ciavarella, and N. Hoffmann, “Reconstruction of governing equations from vibration measurements for geometrically nonlinear systems,” *Lubricants*, vol. 7, no. 8, p. 64, 2019.

[12] Y. Ren, C. Adams, and T. Melz, “Systemidentifikation eines Einmassenschwingers mit spärlicher linearer Regression,” in *DAGA 2020 - 46. Jahrestagung für Akustik*, 2020, pp. 567–570.

[13] G. Box and G. Jenkins, *Time Series Analysis: Forecasting and Control*. Holden-Day, 1970.

[14] G. Dorffner, “Neural Networks for Time Series Processing,” *Neural Network World*, vol. 6, pp. 447–468, 1996.

[15] K. Hornik, M. Stinchcombe, and H. White, “Multilayer feedforward networks are universal approximators,” *Neural Networks*, vol. 2, no. 5, pp. 359–366, 1989.

[16] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[17] Y. Yu, X. Si, C. Hu, and J. Zhang, “A Review of Recurrent Neural Networks: LSTM Cells and Network Architectures,” *Neural computation*, vol. 31, no. 7, pp. 1235–1270, 2019.

[18] F. A. Gers, J. Schmidhuber, and F. Cummins, “Learning to forget: Continual prediction with LSTM,” *Neural computation*, vol. 12, no. 10, pp. 2451–2471, 2000.

[19] R. C. Staudemeyer and E. R. Morris, “Understanding LSTM - a tutorial into Long Short-Term Memory Recurrent Neural Networks,” arXiv, Sep. 2019.

[20] A. Graves, A.-r. Mohamed, and G. Hinton, “Speech Recognition with Deep Recurrent Neural Networks,” arXiv, Mar. 2013.

[21] C. Rackauckas, M. Innes, Y. Ma, J. Bettencourt, L. White, and V. Dixit, “DiffEqFlux.jl - A Julia Library for Neural Differential Equations,” arXiv, Feb. 2019.