Composed Physics- and Data-driven System Identification for Non-autonomous Systems in Control Engineering

Ricarda-Samantha Götte
Heinz Nixdorf Institute
Paderborn University
Paderborn, Germany
rgoette@hni.upb.de

Julia Timmermann
Heinz Nixdorf Institute
Paderborn University
Paderborn, Germany
julia.timmermann@hni.upb.de

Abstract—In control design most control strategies are model-based and require accurate models to be applied successfully. Due to simplifications and the model-reality-gap physics-derived models frequently exhibit deviations from real-world systems. Likewise, purely data-driven methods often do not generalise well enough and may violate physical laws. Recently Physics-Guided Neural Networks (PGNN) and physics-inspired loss functions separately have shown promising results to conquer these drawbacks. In this contribution we extend existing methods towards the identification of non-autonomous systems and propose a combined approach PGNN-L, which uses a PGNN and a physics-inspired loss term (-L) to successfully identify the system’s dynamics, while maintaining the consistency with physical laws. The proposed method is demonstrated on two real-world nonlinear systems and outperforms existing techniques regarding complexity and reliability.

Index Terms—data-driven, physics-based, physics-informed, neural networks, system identification, hybrid modelling

I. INTRODUCTION

In many applied engineering fields, such as control engineering, identifying suitable models is a crucial step to predict a system’s behaviour and be able to control it with a model-based method. Deriving these models by classical scientific approaches, e.g. by formulating the governing equations, preserves the physical principles and therefore enables a physical understanding of the considered system. Nonetheless, these models often exhibit significant deviations from observed quantities due to the model-reality-gap or unmodelled partial dynamics, which cannot be decreased by simple parameter identification techniques.

In recent years this challenge is addressed by taking measurement data into account during the modelling process. This allows a deeper insight into the underlying dynamics [1]. However, data may be limited due to costly experiments, e.g. in automotive test benches, or may be incomplete. Additionally, purely data-driven models lack of physical context. Hence, they may not be capable of describing the system’s dynamics beyond the scope of the used training data. Eventually, this may cause faulty predictions which can harm expensive hardware or increase risks of injury. Thus, currently a field of research on hybrid modelling emerges which seeks to combine physics-based and data-driven approaches towards a more sophisticated modelling [2].

From our point of view there are three perspectives how to use machine learning approaches and physics-based methods or vice versa in conjunction. When computing power had been quite low, physics-based modelling with data-driven aid has been proposed, for instance by extending a physical model with a neural network for parameter estimation [3]. With advancements in storage capacities and computation efficiency, the amount of accessible and processible data increased so that data-driven techniques became popular and physical models rather were utilised for pre-training [4]. Due to the awareness of potential downsides by considering solely one perspective over the other and the acknowledgement of the comprehensive physical knowledge a third perspective arises. Its aim is to assemble physics-based methods and machine learning approaches in a systematic, synergetic way to exploit the domain-specific understanding we usually have and the information we observe due to data [2], [4]. This approach emerges among different scientific disciplines, e.g. in hydrology, climatology, geology, fluid dynamics, and control engineering [4],[6], which illustrates the great promise this field seems to hold for obtaining a deeper understanding of scientific phenomena.

Motivated by the model-reality-gap one of the most popular, yet simple hybrid approaches is Residual Modelling [4], which generally strives to approximate the error between physically derived model’s outputs and system’s measurement data by a data-driven method. Common techniques of modelling the error are least-squares optimisation or neural networks. The advantage of this approach is the simplicity, still it does not enable a deeper insight into possibly unmodelled dynamics, which occur as the deviation between model and measurement, or hold any physical meaning as it just models the deviation from the model to the real-world system. A visualisation of this basic idea is shown in figure 1a. Extending the notion of residual modelling leads to an approach which enhances the simple serial connection of model and data-driven error model and rather takes the error itself directly into account. Recently,
temperatures depending on the depth of a lake based on a

In [2] the authors utilise a PGNN for the estimation of lake

consistency with physical laws [2], [4], [10]–[14].

physical relations, which should guarantee the network’s con-

learning process, e.g. by incorporating bounds or specific

approaches include domain specific knowledge inside the

of physical interpretability. To mitigate this, some research

However, this comes with the drawback of the partial loss

initial values, and a higher likelihood of physical consistency.

by using the simulation model, which is applicable to various

occur. This allows much lower training cost, generalisability

guided by the additional input and mediates when deviations

extent accurate, in the first approach the neural network is

Assuming that the prior physical model is already to somewhat

does not only approximate the error but also learns the relation

between inputs, simulated outputs and measurements as it is

shown in figure [1b]. A very popular technique are Physics-

Guided or Physics-Informed Neural Networks (PGNNs or

PINNs respectively), which consider the simulated model’s

outputs by augmenting the input vector or by using it as

loss function within the learning procedure [2], [4], [7]–[9].

Assuming that the prior physical model is already to somewhat

extent accurate, in the first approach the neural network is

guided by the additional input and mediates when deviations

occur. This allows much lower training cost, generalisability

by using the simulation model, which is applicable to various

initial values, and a higher likelihood of physical consistency.

However, this comes with the drawback of the partial loss

of physical interpretability. To mitigate this, some research

approaches include domain specific knowledge inside the

learning process, e.g. by incorporating bounds or specific

physical relations, which should guarantee the network’s con-

sistency with physical laws [2], [4], [10]–[14].

In [2] the authors utilise a PGNN for the estimation of lake

temperatures depending on the depth of a lake based on a

general lake model. Additionally, they propose a physically

motivated term added to the traditional loss function, which

ensures physical consistency by respecting the relations of

temperature and density as well as density and depth. This

allows the learning procedure to respect physical relations and

provides physically meaningful solutions. In the authors’ work

it can be recognised that this method provides much lower

training error and ensures a higher level of physical correctness

compared to other data-driven methods. Similar to the notion

of incorporating physical relations as an additional loss term,

there are also approaches that exclusively rely on a physical

loss function as error function evaluated on estimated and

targeted outputs, such as the Lagrangian function [12], the

Lagrangian formulation of a system [11], [15] or the governing

equations [6]. If the prior knowledge is accurate enough, these

methods propose good predictions. However, e.g. in case of the

Hamiltonian method, the approach is restricted to conservative

systems which are not likely in practical applications. Moreover,
one needs explicit knowledge about the structure and governing terms of the considered system to

formulate the Lagrangian or the ODEs, which is not realistic

from a practical perspective.

Other hybrid approaches include imposing physics into the

architecture of neural networks as intermediate states [16],

but as stated before these methods require a good knowledge

of the considered system’s structure to be applied successfully.

Moreover, several techniques focus on neural ordinary equa-

tions or even extend that notion towards a neural operator

[17]–[19]. Depending on the considered system, sometimes

a combination of multiple data-driven techniques might be

useful, e.g. the formulation of Bayesian neural network as

PGNN to solve partial differential equations [9].

In this contribution we extend the framework of PGNNs

towards the system identification of non-autonomous systems

and ensure physically consistent predictions by establishing a

physically motivated loss term within the learning procedure.

The proposed approach PGNN-L consists of a PGNN, whose

input is augmented with the control input and the simulated

output of a physical model, which describes the a priori

presumed dynamics and guides the learning process as external

input. This is based on the assumption that the prior model

covers qualitatively the majority of the considered system’s

dynamics, but is not completely reliable in terms of unidenti-

fied parameters and partial dynamics. In contrast to [6],

which utilises the presumed dynamics within the learning procedure,

we impose similar to other researchers [2], [10], [13] loss

functions regarding qualitative information, such as the

energy conservation principle, and investigate the weighting of

multiple loss terms which has been examined very recently for

the first time [20]. The proposed method guarantees a higher

level of physical consistency and outperforms purely physics-

or data-driven methods.

The paper is organised as follows. Section II summarises

classical mathematical or physical modelling approaches and

gives an overview of data-driven methods for identifying

mechatronic systems, whereupon our contribution is presented

Fig. 1: Schemes of approaches for combining data-driven and physics-based models: (a) Residual Modelling. The data-driven model approximates the deviation between the physics-based model’s predictions $y_{phy}$ and the system’s measurements $y$ without diving deeper into the error dynamics. (b) The data-driven model takes the excitation $u$ and the initial state $x_0$ additionally into account, which enables it to recognise the deviations of the physics-based model and utilise these to model the system’s dynamics itself.

(b) Extended Residual Modelling
in detail. In section [III] the hybrid modelling approach is demonstrated and evaluated on two nonlinear systems. The paper concludes with a short discussion of the results in section [IV].

II. MODELLING APPROACHES

A continuous, nonlinear, non-autonomous system is considered

\[
\begin{align*}
\dot{x} &= f(x, u, p), \\
y &= g(x, u, p),
\end{align*}
\]

where \( x \in \mathbb{R}^n \) describes the state, \( u \in \mathbb{R}^m \) the control input, \( y \in \mathbb{R}^l \) the measured output, \( p \in \mathbb{R}^q \) the parameters of the system and \( f : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^q \to \mathbb{R}^n \) and \( g : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^q \to \mathbb{R}^l \) are nonlinear functions, describing the system’s dynamics.

In control engineering, we wish to model the plant and identify the parameters \( p \) to proceed with the model-based control design, including the synthesis, analysis and simulation of each design step, e.g. the establishment of a controller. In this work we focus on the aforementioned first step of control design, the system and parameter identification.

A. Mathematical Modelling

Traditionally, real-world dynamics are characterised by a set of governing equations, derived from physical laws. Depending on the considered system, either Newton’s axioms or Euler-Lagrange formalism are utilised for formulating continuous, potentially nonlinear models which describe the system’s dynamics by ordinary differential equations (ODEs). In this work we assume that the considered system can be captured by the following system of ODEs

\[
\begin{align}
\dot{x}_{phy} &= f_{phy}(x_{phy}, u, \tilde{p}), \\
y_{phy} &= g_{phy}(x_{phy}, u, \tilde{p}),
\end{align}
\]

in a qualitative manner which means it is not as accurate as we wish due to possibly unidentified parameters \( \tilde{p} \in \mathbb{R}^q \) or unmodelled partial dynamics. This may be the case e.g. during the commissioning of a plant or when complex dynamical elements are neglected or unknown.

In many fields, not only in control engineering, this model often is too complex in terms of dimension or nonlinearity to be utilised for any method, so that it needs to be simplified by linearisation techniques, such as Taylor expansion within concrete operating points. These models then can be applied in model-based methods, which require a fast and simple evaluation of the model to be able to predict, for instance model-predictive control, gain scheduling approaches and similar methods. The disadvantage of those derived models is primarily the expense to deduce and store a model for each operating point, but more important the potential of errors due to the simplified model. Therefore, it is always desirable to obtain a good trade-off between a model’s accuracy and its complexity.

B. Data-driven Modelling

Lately, the utilisation of data in science and engineering becomes increasingly popular. In modelling complex processes, data-driven techniques may offer the opportunity to optimise the process and save resources like time, energy or manpower, for instance throughout the set-up and deployment of a novel plant. During regular operation but also even before, when installing and testing a novel plant by applying small excitations, data \( D \subset \mathbb{R}^{(l+m) \times N} \) can be collected by \( N \in \mathbb{N} \) measurement samples of the plant’s excitation and output, \( u_{M,k} \) and \( y_{M,k} \) respectively, and structured time \( t(k) = t_k \) dependently in the following way

\[
D = \{(y(t_k), u(t_k))\} = \{(y_{M,k}, u_{M,k})\} \quad k = 0, 1, \ldots, N.
\]

Thereby, the samples need not necessarily belong to one trajectory, rather should cover various states and excitations in the phase and input space except if the plant is in the process of deployment. Using these samples, different methods can be applied to derive a data-driven model which approximates the system to the best possible accuracy.

Least-squares optimisation is a very popular approach among machine learning techniques, which attempts to fit measurement data to a certain model, e.g. a linear regression model, by reducing the least-squares error [21].

A similar approach, utilising least-squares optimisation, is called Sparse Identification of Nonlinear Dynamics with control (SINDYc) and based on the assumption that many systems’ dynamics can be described by only a small number of governing terms [1]. Organising the measurement data \( D \), derived from the system, into snapshot matrices

\[
Y = \begin{pmatrix}
y_{M,1} & y_{M,2} & \cdots & y_{M,N-1}
y_{M,2} & y_{M,3} & \cdots & y_{M,N}
y_{M,3} & y_{M,4} & \cdots & \vdots
\end{pmatrix} \in \mathbb{R}^{l \times (N-1)},
\]

\[
Y' = \begin{pmatrix}
y_{M,2} & y_{M,3} & \cdots & y_{M,N}
y_{M,3} & y_{M,4} & \cdots & \vdots
\end{pmatrix} \in \mathbb{R}^{l \times (N-1)},
\]

\[
U = \begin{pmatrix}
u_{M,1} & u_{M,2} & \cdots & u_{M,N-1}
u_{M,2} & u_{M,3} & \cdots & \vdots
\end{pmatrix} \in \mathbb{R}^{m \times (N-1)},
\]

the aim is to find a best-fit approximation model \( Y' = \Xi \Psi(Y, U)^T \) by storing and evaluating \( \kappa \) initial guess terms in a library vector \( \Psi(Y, U) \in \mathbb{R}^{(N-1) \times \kappa} \), which might describe the considered system’s dynamics well. The method then seeks a sparse vector \( \Xi \in \mathbb{R}^{l \times \kappa} \) such that the following holds for each row \( i \) with \( 0 < \lambda \ll 1 \) as a model complexity parameter

\[
\xi_i^* = \arg\min_{\xi_i} \frac{1}{2} \left\| Y_i' - \xi_i \Psi(Y, U)^T \right\|^2_2 + \lambda |\xi_i|. \tag{5}
\]

This technique is usually considered as data-driven method as it can be utilised without any prior knowledge. However, if prior knowledge is available, SINDYc moves towards the perspective of hybrid modelling approaches by incorporating preassumed terms into the library vector. Advantages of
SINDYc include the computational efficiency, its real-time capability, once $\Xi$ is determined, and the preservation of the system’s physical interpretability by $\Xi$, indicating which terms are relevant for the description of system (1). Nonetheless, it relies on the assumption of a few governing terms, which is a huge drawback because it might not be true for many complex systems. Furthermore, its model is not capable of imposing any physical or operational limitations, restricting its applicability in many mechatronic systems.

**Neural Networks (NNs)** have become very successful during the last years in a variety of applications, e.g. learning control policies [22] or addressing the problem of non-measurable states [23]. Data are fed into input layers, processed through a number of hidden layers with (nonlinear) activation functions and transformed to estimated outputs $\hat{Y}$, which are compared to labelled data $Y$ in the following way [21]:

$$L_{\text{error}}(Y, \hat{Y}) = \frac{1}{N} \sum_{k=1}^{N} \left| \left| Y_k - \hat{Y}_k \right| \right|_2^2.$$  (6)

Depending on the shape of data, other norms than the mean squared error may be applied equally, e.g. the mean absolute error. However, the choice of the norm for the loss function (6) as well as the choice of activation functions and the number of neurons and layers strongly affect the network’s performance and therefore should be chosen carefully and problem-specifically. Due to these numerous hyperparameters, influencing directly the training success, it is necessary to find the best suitable set of hyperparameters. Among different methods, e.g. random search or grid search, Bayesian optimisation has become the state-of- the-art technique to determine optimal hyperparameters for NNs. Eventually the error, resulting from the comparison (6), is processed via backpropagation and used to adapt the neurons for a better fit [21]. The capability of a NN as a global approximator of any function results from its error itself (Residual Modelling). As NNs often tend to overfitting, an obligatory step of data preprocessing is the standardisation of data. Therefore, before using the excitation, simulated output or measurement data as inputs for the NN, this step has to be applied towards each of them. If data feature different scales, normalisation or a logarithmic transformation of the samples may be considered additionally.

In contrast to [6], [8], [11], [12] we take the prior physical knowledge as external input into account which rather guides the PGNN than forces it to strictly obey a priori presumed dynamics. Using such models of presumed dynamics within the loss function would imply we consider the model (2) to be completely true. Due to the model-reality-gap this is hard to achieve and would lead the PGNN to learn possibly faulty dynamics. Subsequently, the PGNN would not yield a significant higher model accuracy than the presumed model and justify the effort of hybrid modelling or the loss of interpretability. Hence, it is more efficient to let the model [2] guide the PGNN’s learning process externally with a higher flexibility of adopting or neglecting information. However, data-driven methods in general and NNs in particular, have significant drawbacks that result in poor solutions for unseen
During training the PGNN(-L) takes the excitation $u^k$, the simulated output $(x_{phy}^k, y_{phy}^k)$ of the discrete physical model and the measurement data $y^k$ into account to minimise the error $e_{PGNN}$ of its loss function (6) (or in case of PGNN-L (7)).

Once the PGNN(-L) is trained, the prediction of the previous time step $\hat{y}^k$ is constantly fed back into its input layer. With the current excitation $u^k$ and simulated output $(x_{phy}^k, y_{phy}^k)$ a prediction for the next time step can be estimated.
for prediction as it is illustrated in figure 2b. It shows the prediction stage as an integration process which needs the current excitation \( u^k \) as only information from the plant, the prediction \( \hat{y}^k \) of the previous time step, which is constantly fed back into the NN as initial value, and the estimated prediction of the discrete physical model of the current time step, which is equally fed back into the physical model as initial value for the next time step. Provided the predictions of the simulation model \( \hat{X} \) are already calculated and the NN features a low complexity, it is very efficient and fast to evaluate the PGNN-L due to the computation of nonlinear matrix-vector-computations. However, even with simulating the physical model beforehand, it may be capable for real-time applications, depending on the number of neurons and layers as well as on the complexity of the physical model.

III. EXPERIMENTAL RESULTS

The previous described method is eventually demonstrated on two non-autonomous, nonlinear systems. From now on we assume that the collected data samples include the full state of the considered system \( (y = x) \), which is reasonable as an initial observer can be built upon prior knowledge of the physical model \( \hat{X} \). This might lead to slightly corrupted data but can be compensated by the hybrid approach. For each considered system the following modelling approaches are compared throughout the paper: The typical physics-based model description, a solely data-driven, feed-forward neural network (NN), a PGNN with the same inner structure like the NN and a physical loss term (PGNN-L) as well as a SINDYc model based on the prior knowledge the physical model contains.

A. Golf robot

The first system to be considered is a golf robot, a test rig at our laboratory to investigate machine learning based methods for hitting a golf ball within the green, which is illustrated in figure 3a. The physical model is captured by a nonlinear, input affine system of ODEs:

\[
\dot{x} = \begin{pmatrix} -mg_a \sin x_1 - F_G(x) + 4u_G \end{pmatrix}, \quad F_G(x) = dx_2 + r \mu \text{sign}(x_2) \left( mx_2^2a + mg \cos x_1 \right). \tag{9}
\]

The motor torque, which is multiplied by four due to the gear ratio, serves as control input \( u_G \in \mathbb{R} \), while the angle \( \varphi \) and angular velocity \( \dot{\varphi} \) of the club head describe the golf robot’s states \( x \in \mathbb{R}^2 \). As stated before, only \( \varphi \) is measured, whereas \( \dot{\varphi} \) is estimated. The term \( F_G(x) \) contains the dynamics resulting from static and sliding friction. The parameters \( \tilde{p} = (m, a, J, d, r, \mu)^T \in \mathbb{R}^6 \), estimated by measuring and a particle swarm optimisation, are shown in table 3a. Due to the stick-slip-effect the simulation of the model \( \hat{X} \) exhibits significant deviations compared to measurements from the test rig, which is illustrated in figure 4 by the green dashed and black solid lines, respectively. To overcome this discrepancy, a PGNN-L with a simple feed-forward architecture, including two hidden layers, is trained by minimising the following specific form of the loss function \( L \):

\[
L(\hat{X}, X, U) = (1 - \lambda_{phys}) \frac{1}{N} \sum_{k=1}^{N} (\hat{X}_k - X_k)^2 + \lambda_{phys} \frac{1}{N-1} \sum_{k=2}^{N} \Delta E(\hat{X}_k, \dot{\hat{X}}_{k-1}, U_{k-1})^2
\]

\[
\Delta E(\hat{X}_k, \dot{\hat{X}}_{k-1}, U_{k-1}) = \Delta E_{\text{kin}}(\hat{X}_k, \dot{\hat{X}}_{k-1}) + \Delta E_{\text{pot}}(\hat{X}_k, \dot{\hat{X}}_{k-1}) + \Delta E_{\text{con}}(\hat{X}_k, \dot{\hat{X}}_{k-1}, U_{k-1}) - \Delta E_{\text{diss}}(\hat{X}_k, \dot{\hat{X}}_{k-1}), \tag{10}
\]

for which the indices under each \( \Delta E \) mark the kinetic, potential, control supplied and dissipative energy fractions, respectively. In contrast to [13], the second loss term extends the scope of conservative systems towards a variety of systems by taking additionally the dissipative and control supplied energy fractions into account. Hence, the second term in the loss function (10) measures the energy balance between the previous and the current time step, which would be zero if the system’s energy balance is modelled correctly. However, recalling that the prior knowledge and thus the energy balance are incomplete due to unmodelled partial dynamics, the physical loss term is independent of labelled data and is only evaluated on predicted outputs. In [13] the authors compare the conservative system’s energy balance between targeted and predicted outputs to ensure physical consistency. This is reasonable if the system is completely known. Yet, in this work this is not the case so the loss term evaluated on predicted outputs ensures the learning problem shifts to the direction of physically consistent solutions.

Data for training include measurement samples from six sinusoidal-, step- and chirp-excited trajectories with a sampling rate of 1 kHz. These are separated in 60-20-20 percentage fractions for training, validation and test data sets. For both, the NN and the PGNN-L, a hyperparameter optimisation is performed by Bayesian optimisation to obtain the best set of affecting parameters, including \( \lambda_{phys} \) for the PGNN-

| Parameter                      | Value   |
|--------------------------------|---------|
| mass of club head \( m \)      | 0.5241 kg |
| length from center of gravity to rotation axis \( a \) | 0.4702 m |
| moment of inertia \( J \)       | 0.1445 kgm² |
| damping constant \( d \)        | 0.0132 kgs⁻¹ |
| distance friction point to rotation axis \( r \) | 0.0245 m |
| friction coefficient \( \mu \)  | 1.5136 |
| gravity \( g \)                | 9.81 ms⁻² |

(a) Construction  
(b) Parameters \( \tilde{p} \)

Fig. 3: Golf robot
Fig. 4: Prediction of golf robot’s dynamics with $\lambda_{phy} = 0.8175$, number of PGNN’s and NN’s neurons: 27 vs. 39

Fig. 5: Effects of the training data used: If only a small data amount covering the transient process is used, the SINDYc model will fail to reproduce the golf robot’s angle due to missing information within the data. In contrast, the PGNN-L will capture the dynamics because of its additional information from the prior model and physical loss term.

L. The SINDYc library vector $\Psi$ contains the states, the control input as well as trigonometric functions to find a suitable $\Xi$. Eventually, a reference trajectory from the test rig is used for simulation to quantify the models’ accuracy, which is evaluated between the measured angle $\varphi^k$ and the corresponding prediction $\hat{\varphi}^k$.

Figure [4] shows the resulting trajectories of each considered method. It reveals that the PGNN-L is confident to capture the golf robot’s dynamics to a higher accuracy than the prior model. This is verified by the error displayed in table II. Although it covers the dynamics already very closely and does not exhibit significant phase shifts, it sometimes fails
to reproduce the exact amplitude. However, it outperforms a
traditional NN clearly, which is even worse than the physical
model. For instance, the NN is not capable to recognise the
friction and still oscillates a long time after the system has
reached its equilibrium. Furthermore, the NN’s complexity
is much higher than the PGNN-L’s, which only needs 27
neurons. Compared to the SINDYc model, the PGNN-L’s
accuracy appears to be slightly worse which is confirmed
by the error in table [I]. Nevertheless, if training data only
covers a small region of the phase and input space, e.g. the
transient process of each of the previous for training used
six trajectories, the PGNN-L approach clearly is superior over
SINDYc. This is illustrated in figure 5 whose top figure shows
simulated trajectories of the angle after training with data
that covers a reasonable amount of the phase and input space
and whose medium figure displays results of the angle after
training with data that covers only the transient process. In this
situation the advantage of the PGNN-L becomes clear: Since
it relies on the physical model and a qualitative loss term, it
can cope with incomplete data in contrast to SINDYc which
approximates the dynamics poorly. Due to its reliance on data
and a few governing terms SINDYc needs representative data
of the entire phase and input space to approximate a confident
model. Therefore, if only incomplete data combined with some
physical understanding is available, which is often the case in
complex industrial applications, PGNN-L is the optimal choice
to derive a model.

Besides the PGNN-L’s accuracy the trade-off between the
competing loss terms $L_{\text{error}}$ and $L_{\text{phy}}$ during training is
examined. In figure 6 the pareto front is shown for varying
weighting factors $\lambda_{\text{phy}}$. If $\lambda_{\text{phy}}$ is 0.99, which means the
focus on the data-fit term (6) is very low, the error between
targeted and predicted angle is high. However, zooming into
the rectangle reveals a typical convex form of a pareto front.
It indicates that a good weighting factor for the golf robot is
between 0.8 and 0.9 meaning the two competing loss terms
are both very small. This is a high emphasis on the energy
conservation loss term. Although the PGNN-L model is at
least as good as the SINDYc model or in case of low data
availability even more precise, simulations as in figure 4 and
figure 5 still exhibit some deviations between the PGNN-L
model and the true dynamics, e.g. in the amplitudes. Therefore,
a better fit towards the data would be desirable. Instead of
setting $\lambda_{\text{phy}}$ constant overall the learning process, $\lambda_{\text{phy}}$ may
be regarded as an adaptive parameter similar to the learning
rate which is able to vary according to the training loss. Future
work will examine this strategy.

B. Servo valve

In many hydraulic applications servo valves are frequently
utilised as actuators. However, servo valves often feature
complex nonlinear dynamics, for instance the bandwidth of the
valve strongly depends on the amplitude of the input signal.
Therefore, it is common to utilise simplified models for valve
dynamics. An approach with very low modelling depth is e.g.
the description of the servo valve dynamics via a second order
system:

$$\dot{x} = \left( -\frac{2D_v}{T_v} x_2 + \frac{x_2}{T_v} x_1 + \frac{K_v}{T_v} u_V \right), \quad (11)$$

with $x \in \mathbb{R}^2$ describing the position $y_V$ and the velocity $\dot{y}_V$
of the valve slider, $u_V \in \mathbb{R}$ representing the voltage the servo
valve receives as input signal and $T_v, D_v$ and $K_v$ denoting
the typical parameters of a second order system. Again, only
$y_V$ is measured, whereas $\dot{y}_V$ is estimated. In this work we
consider a four way three position servo valve consisting of a
valve slider, a torque motor and a hydraulic amplifier based
on the nozzle-baffle principle which is illustrated in figure 7.

![Fig. 7: Servo valve deployed in a test rig](image)

The valve slider is controlled with electric position control. The
servo valve’s parameters for the physical model (11) are displayed in table II. Although the model (11) is able to predict the system’s behaviour in general, it fails to preserve
the intrinsic limitations of the valve slider’s velocity and
acceleration due to the oil viscosity inside the piston and
due to the torque motor’s position because of its operational
PGNN-L with prior model that respects physical limitations

The library vector for SINDYc contains the states determined by Bayesian optimisation, including \( \lambda \) both NN and PGNN-L the best set of hyperparameters is.

Methods are able to achieve. Although the physical model complexity. One would expect the more precise the prior model is the better the PGNN-L's approximation of the true dynamics will be. This needs to be examined further in future work. Nonetheless, the PGNN-L is less complex than a standard NN and features only an eighth of the NN's number of neurons remains constant which need not necessarily mean the modelling depth does not influence the PGNN-L's model complexity. One would expect the more precise the prior model is the better the PGNN-L's approximation of the true dynamics will be. This needs to be examined further in future work. Nonetheless, the PGNN-L is less complex than a standard NN and features only an eighth of the NN's number of neurons. This benefit is caused by the utilisation of the prior model and the physically motivated loss term. Moreover, it is remarkable that the SINDYc model is not capable to recognise any intrinsic bounds and therefore reproduces the valve's dynamics very poorly. Since SINDYc operates by providing governing terms, it cannot respect any limitations. Thus, the PGNN-L is the optimal choice to cover the valve's real-world dynamics among the examined techniques.

**TABLE I: Parameters \( \tilde{\rho} \)**

| Parameter          | Value     |
|--------------------|-----------|
| natural frequency  | 350 Hz    |
| damping ratio      | 0.5       |
| gain factor        | 0.1       |
| voltage \( u_V \)  | -10 \( \ldots \) 10 V |
| maximal position   | 4.2672 \( \times 10^{-4} \) m |
| supply pressure    | 280 bar   |

Fig. 8: Prediction of servo valve dynamics by a step excitation of 5V at 1s: \( 8a \) \( \lambda_{phy} = 0.2527 \), number of PGNN-L's and NN's neurons: 11 vs. 92, \( 8b \) \( \lambda_{phy} = 0.3206 \), number of PGNN-L's and NN's neurons: 11 vs. 92

- Oscillating and damping effects. If the physical model (11) is improved by estimated bounds of the slider's velocity and acceleration due to the oil's viscosity, this information can be preserved by the PGNN-L according to figure [8a]. Here, the PGNN-L reproduces the valve’s real-world dynamics even more precisely compared to the setting before. A much lower error in table [II] verifies this observation. Hence, the modelling depth of the prior model appears to directly affect the PGNN-L's performance and indicates a specific modelling depth is essential for deriving meaningful hybrid models. However, the number of neurons remains constant which need not necessarily mean the modelling depth does not influence the PGNN-L's model complexity.

**TABLE II: Root-mean-squared error (RMSE) of the considered nonlinear systems**

| Method       | Golf robot \( [\text{fig. 4}] \) | Valve \( [\text{fig. 8a}] \) | Valve \( [\text{fig. 8b}] \) |
|--------------|---------------------------------|----------------------------|----------------------------|
| PhysModel    | 6.6563 \( \times 10^{-4} \)     | 5.0501 \( \times 10^{-4} \) | 3.9959 \( \times 10^{-4} \) |
| NN           | 8.9736 \( \times 10^{-4} \)     | 2.1382 \( \times 10^{-4} \) | 2.1382 \( \times 10^{-4} \) |
| SINDYc       | 3.4516 \( \times 10^{-4} \)     | 2.8619 \( \times 10^{-4} \) | 2.8619 \( \times 10^{-4} \) |
| PGNN-L       | 3.7982 \( \times 10^{-4} \)     | 1.1982 \( \times 10^{-4} \) | 7.3835 \( \times 10^{-5} \) |

- This is illustrated in figure [8a] by the green dotted and the black solid lines, respectively. Analogue to the first considered system, a PGNN-L with feed-forward structure and two hidden layers is trained via the loss function (10). Data for training include ten, slightly noisy trajectories of different step excitations \( u_V \) with a sampling rate of 2 kHz and are separated in the same manner (60-20-20) for training, validation and test data sets as in the previous section. Furthermore, for both NN and PGNN-L the best set of hyperparameters is determined by Bayesian optimisation, including \( \lambda_{phy} \) for the PGNN-L. The library vector for SINDYc contains the states \( y_V, \dot{y}_V \) and the voltage \( u_V \) within this example. Each model's accuracy is quantified by the same reference trajectory from measurement data, which has not been covered during training, and evaluated between the measured state \( y_V^k \) and the predicted state \( \hat{y}_V^k \).

- Figure [8b] shows that the PGNN-L is able to predict the servo valve’s behaviour to a higher accuracy than the compared methods are able to achieve. Although the physical model reveals significant deviations especially in the magnitude, e.g. in the slider’s velocity, the PGNN-L can process this information and utilize it with data and the energy respecting loss term to capture the true servo valve’s dynamics closely. Neither the standard NN nor the SINDYc model can reproduce the dynamics to this accuracy. Both fail in recognising the

- TABLE I: Parameters \( \tilde{\rho} \)

- TABLE II: Root-mean-squared error (RMSE) of the considered nonlinear systems

- Oscillating and damping effects. If the physical model (11) is improved by estimated bounds of the slider’s velocity and acceleration due to the oil’s viscosity, this information can be preserved by the PGNN-L according to figure [8a]. Here, the PGNN-L reproduces the valve’s real-world dynamics even more precisely compared to the setting before. A much lower error in table [II] verifies this observation. Hence, the modelling depth of the prior model appears to directly affect the PGNN-L's performance and indicates a specific modelling depth is essential for deriving meaningful hybrid models. However, the number of neurons remains constant which need not necessarily mean the modelling depth does not influence the PGNN-L's model complexity. One would expect the more precise the prior model is the better the PGNN-L’s approximation of the true dynamics will be. This needs to be examined further in future work. Nonetheless, the PGNN-L is less complex than a standard NN and features only an eighth of the NN’s number of neurons. This benefit is caused by the utilisation of the prior model and the physically motivated loss term. Moreover, it is remarkable that the SINDYc model is not capable to recognise any intrinsic bounds and therefore reproduces the valve’s dynamics very poorly. Since SINDYc operates by providing governing terms, it cannot respect any limitations. Thus, the PGNN-L is the optimal choice to cover the valve’s real-world dynamics among the examined techniques.

- TABLE I: Parameters \( \tilde{\rho} \)

- TABLE II: Root-mean-squared error (RMSE) of the considered nonlinear systems

- Oscillating and damping effects. If the physical model (11) is improved by estimated bounds of the slider’s velocity and acceleration due to the oil’s viscosity, this information can be preserved by the PGNN-L according to figure [8a]. Here, the PGNN-L reproduces the valve’s real-world dynamics even more precisely compared to the setting before. A much lower error in table [II] verifies this observation. Hence, the modelling depth of the prior model appears to directly affect the PGNN-L's performance and indicates a specific modelling depth is essential for deriving meaningful hybrid models. However, the number of neurons remains constant which need not necessarily mean the modelling depth does not influence the PGNN-L's model complexity. One would expect the more precise the prior model is the better the PGNN-L’s approximation of the true dynamics will be. This needs to be examined further in future work. Nonetheless, the PGNN-L is less complex than a standard NN and features only an eighth of the NN’s number of neurons. This benefit is caused by the utilisation of the prior model and the physically motivated loss term. Moreover, it is remarkable that the SINDYc model is not capable to recognise any intrinsic bounds and therefore reproduces the valve’s dynamics very poorly. Since SINDYc operates by providing governing terms, it cannot respect any limitations. Thus, the PGNN-L is the optimal choice to cover the valve’s real-world dynamics among the examined techniques.

- Oscillating and damping effects. If the physical model (11) is improved by estimated bounds of the slider’s velocity and acceleration due to the oil’s viscosity, this information can be preserved by the PGNN-L according to figure [8a]. Here, the PGNN-L reproduces the valve’s real-world dynamics even more precisely compared to the setting before. A much lower error in table [II] verifies this observation. Hence, the modelling depth of the prior model appears to directly affect the PGNN-L's performance and indicates a specific modelling depth is essential for deriving meaningful hybrid models. However, the number of neurons remains constant which need not necessarily mean the modelling depth does not influence the PGNN-L's model complexity. One would expect the more precise the prior model is the better the PGNN-L’s approximation of the true dynamics will be. This needs to be examined further in future work. Nonetheless, the PGNN-L is less complex than a standard NN and features only an eighth of the NN’s number of neurons. This benefit is caused by the utilisation of the prior model and the physically motivated loss term. Moreover, it is remarkable that the SINDYc model is not capable to recognise any intrinsic bounds and therefore reproduces the valve’s dynamics very poorly. Since SINDYc operates by providing governing terms, it cannot respect any limitations. Thus, the PGNN-L is the optimal choice to cover the valve’s real-world dynamics among the examined techniques.
IV. CONCLUSION AND OUTLOOK

In this contribution we extend the framework of PGNNs towards system identification of non-autonomous systems in control engineering. The described approach PGNN-L consists of a PGNN with augmented inputs by the control input and simulated output of the corresponding prior model as well as of a physics-motivated loss term within the learning procedure. Two real-world nonlinear systems are examined, for which the physical loss term respects the energy conservation principle and takes also dissipative and control supplied energy fractions into account. It is shown that the method achieves a higher model accuracy than existing methods, such as NN or SINDYc models, if incomplete or noisy data is available. It also respects physical limitations. Additionally, it enables mostly lower model complexity, since it is guided by the prior physical model. Furthermore, its physics-motivated loss term ensures physically consistent predictions to a higher extent, which is considered as robustness criterion. Future work include the investigation of balancing more competing physical loss terms, including vague knowledge, the effects of different modelling depths for the prior model and the development of an adaptive weighting factor schedule for the physics-based loss term.

ACKNOWLEDGMENT

This work was developed in the junior research group DART (Datengetriebene Methoden in der Regelungstechnik), University Paderborn, and funded by the Federal Ministry of Education and Research of Germany (BMBF - Bundesministerium für Bildung und Forschung) under the funding code 01IS20052. The responsibility for the content of this publication lies with the authors.

REFERENCES

[1] S. L. Brunton, J. L. Proctor, and J. N. Kutz, “Sparse identification of nonlinear dynamics with control (synsic),” in IFAC-PapersOnLine, 2016, vol. 49, pp. 710–715.
[2] A. Karpatne, G. Alur, J. Faghmous, M. Steinbach, A. Banerjee, A. Ganguly, S. Shekhar, N. Samatova, and V. Kumar, “Theory-guided data science: A new paradigm for scientific discovery from data,” in IEEE Transactions on Knowledge and Data Engineering, 2017, vol. 29, pp. 2318–2331. [Online]. Available: http://arxiv.org/pdf/1612.08543v2
[3] D. C. Psichogios and L. H. Ungar, “A hybrid neural network-first principles approach to process modeling,” in AIChE Journal, 1995, vol. 38.
[4] J. Willard, X. Jia, S. Xu, M. Steinbach, and V. Kumar, “Integrating physics-based modeling with machine learning: A survey,” 2020. [Online]. Available: http://arxiv.org/pdf/2003.04919v4
[5] A. Karpatne, W. Watkins, J. Read, and V. Kumar, “Physics-guided neural networks (pgnn): An application in lake temperature modeling,” 2017. [Online]. Available: http://arxiv.org/pdf/1710.11431v2
[6] E. A. Antonelo, E. Camponogara, L. O. Seman, E. R. de Souza, J. P. Jordanau, and J. F. Hübner, “Physics-informed neural nets-based control,” 2021. [Online]. Available: http://arxiv.org/pdf/2004.01001v6
[7] J. Pathak, A. Wikner, R. Fussell, S. Chandra, B. R. Hunt, M. Girvan, and E. Ott, “Hybrid forecasting of chaotic processes: Using machine learning in conjunction with a knowledge-based model,” in Chaos, 2018, vol. 28, p. 041101.
[8] Y. Yu, H. Yao, and Y. Liu, “Structural dynamics simulation using a novel physics-guided machine learning method,” in Engineering Applications of Artificial Intelligence, vol. 96, 2020, p. 103947.
[9] L. Yang, X. Meng, and G. E. Karniadakis, “B-pins: Bayesian physics-informed neural networks for forward and inverse pde problems with noisy data,” 2020. [Online]. Available: http://arxiv.org/pdf/2003.06097v1
[10] N. Muradligh, M. R. Islam, M. Marwah, A. Karpatne, and N. Ramakrishnan, “Incorporating prior domain knowledge into deep neural networks,” in 2018 IEEE International Conference on Big Data, 2018, pp. 36–45.
[11] M. Cramer, S. Greydanus, S. Hoyer, P. Battaglia, D. Sperguson, and S. Ho, “Lagrangian neural networks,” 2020. [Online]. Available: http://arxiv.org/pdf/2003.04630v2
[12] S. Greydanus, M. Dzamba, and J. Yosinski, “Hamiltonian neural networks,” 2019. [Online]. Available: http://arxiv.org/pdf/1906.01563v3
[13] S. J. Raymond and D. B. Camarillo, “Applying physics-based loss functions to neural networks for improved generalizability in mechanics problems,” in Journal of Computational Physics, 2021. [Online]. Available: http://arxiv.org/pdf/2105.00074v1
[14] L. von Rueden, S. Mayer, K. Beckli, B. Georgiev, S. Giesselbach, R. Reese, B. Kirsch, J. Pfommer, A. Pick, R. Ramamurthy, M. Walczak, J. Garcke, C. Bauchhage, and J. Schucker, “Informed machine learning – a taxonomy and survey of integrating knowledge into learning systems,” 2019. [Online]. Available: http://arxiv.org/pdf/1903.12394v2
[15] M. Lutter, C. Ritter, and J. Peters, “Deep lagrangian networks: Using physics as model prior for deep learning,” in ICLR, 2015. [Online]. Available: http://arxiv.org/pdf/1907.04490v1
[16] A. Daw, R. Q. Thomas, E. C. Carey, J. S. Read, A. P. Applling, and A. Karpatne, “Physics-guided architecture (pga) of neural networks for quantifying uncertainty in lake temperature modeling,” 2019. [Online]. Available: http://arxiv.org/pdf/1911.02682v1
[17] B. Chen, H. Zhang, X. Liu, and C. Lin, “Neural observer and adaptive neural control design for a class of nonlinear systems,” in IEEE Transactions on Neural Networks and Learning Systems, 2018, vol. 29, pp. 4261–4271.
[18] Z. Li, N. Kovachki, K. Azizzadenesheli, B. Liu, K. Bhattacharya, A. Stuart, and A. Anandkumar, “Neural operator: Graph kernel network for partial differential equations,” 2020. [Online]. Available: http://arxiv.org/pdf/2003.03485v1
[19] ——, “Fourier neural operator for parametric partial differential equations,” in ICLR, 2021. [Online]. Available: http://arxiv.org/pdf/2010.08895v2
[20] M. Rohrofer, S. Posch, and B. C. Geiger, “On the pareto front of physics-informed neural networks,” 2021. [Online]. Available: http://arxiv.org/pdf/2012.04852v3
[21] C. M. Bishop, Pattern recognition and machine learning, ser. Information science and statistics. Springer, 2006.
[22] Y.-C. Chang, N. Roohi, and S. Gao, “Neural lyapunov control,” 2019. [Online]. Available: https://arxiv.org/abs/2005.00611
[23] R. T. Q. Chen, Y. Rubanova, J. Bettencourt, and D. Duvenaud, “Neural ordinary differential equations,” in 32nd Conference on Neural Information Processing Systems (NeurIPS), 2018. [Online]. Available: http://arxiv.org/pdf/1806.07366v2
[24] T. Chen and H. Chen, “Approximation capability to functions of several variables, nonlinear functionals, and operators by radial basis function neural networks,” in IEEE Transactions on Neural Networks, 1995, vol. 6, pp. 904–910.
[25] M. Raissi, P. Perdikaris, and G. E. Karniadakis, “Physics-informed neural networks,” 2019. [Online]. Available: http://arxiv.org/pdf/1906.15286v2
[26] M. Lutter, C. Ritter, and J. Peters, “Deep lagrangian networks: Using physics as model prior for deep learning,” in ICLR, 2015. [Online]. Available: http://arxiv.org/pdf/1903.12394v2
[27] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” in ICLR, 2015. [Online]. Available: http://arxiv.org/pdf/1412.6980v9
[28] B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, and N. de Freitas, “Taking the human out of the loop: A review of bayesian optimization,” in Proceedings of the IEEE, 2016, vol. 104, pp. 148–175.