Neural Network Augmented Physics Models for Systems with Partially Unknown Dynamics: Application to Slider-Crank Mechanism

Wannes De Groote, Edward Kikken, Erik Hostens, Sofie Van Hoecke, Guillaume Crevecoeur

Abstract—Mechatronic systems are plagued by nonlinearities and contain uncertainties due to, amongst others, interactions with their environment. Models exhibiting accurate multistep predictive capabilities can be valuable in the context of motion control and design of servo controlled systems. Neural Network Augmented Physics (NNAP) models are presented in this paper to comprehend the behavior of servo systems that contain partially unknown dynamics. By means of a hybrid modeling formalism, neural network models are closely merged with physics-based state-space derivative functions. The methodology is applied on an experimental slider-crank system consisting of a servo drive and mechanical links of which the physical behavior is only partially known. Its interaction with the environment due to a compression spring load and friction phenomena, however, remains unknown. Accurate prediction capabilities of the presented NNAP models are demonstrated on the slider-crank system. Compared to the feedforward modeling formalism, recurrent NNAP models had the ability to even further reduce prediction errors up to 43.3% on validation data. From the converged dynamic NNAP model we extracted the neural network and identified the unknown phenomena, being the spring characteristic and the friction forces, within the mechatronic system.

Index Terms—Nonlinear Dynamic System Modeling, Multistep Prediction, Neural Networks, Explainable Artificial Intelligence, Load Identification

I. INTRODUCTION

SERVO controlled systems face increasingly demanding performance and efficiency demands in industrial and manufacturing applications. In amongst others presses, pumps and compressors, linear motion needs to be realized in direct drive or indirectly. The latter servo systems need to drive a load system via a rotary motor system and mechanical transmissions such as gears, cam-follower and bar linkages. Position and speed control are critical and their performance is limited due to unknown dynamics resulting from the external load and uncertainties in the mechanical system such as inertia, backlash, damping and friction [1]–[3]. To better understand these intricate and often highly nonlinear dynamics, extensive research is conducted on developing prediction models for capturing the overall behavior of servo systems that can be inserted in control strategies, as in [4]. These predictive models can furthermore enable a user or manufacturer to further improve the mechatronic system design [5], [6].

Capturing the servo drive system dynamics is however often cumbersome and challenging as we come up against limits of various existing modeling and system identification formalisms. The modeling of a mechatronic system can be based on expert knowledge where partial and/or ordinary differential equations can be distilled from the system. These physics-based models include (lumped) physical parameters that can be directly related to the actual mechatronic system. Parameter identification techniques can then be used to identify the values of these parameters based on experimental sensor data by aligning the model response with the data [7]. Uncertainties can however still remain as servo drive systems face nonlinearities such as friction forces that give rise to disturbances that are difficult to model. Alternatively, black-box models can be built in a data-driven manner. In the context of nonlinear system identification several methodologies exist such as Gaussian processes [8], Hammerstein-Wiener structures [9] and nonlinear auto-regressive models (e.g. NARMAX) [10]. Supervised learning is typically used to build the model by relating input to output data. Supervised machine learning techniques such as deep learning have growing influence as they have the ability to recognize patterns in data [11]–[13]. These so-called neural networks have been used to approximate various nonlinear relations such as inverse kinematics within complex mechatronic systems [14]. Recurrent neural networks (RNN) have the ability to capture the nonlinear dynamics in time series by means of an internal state and is of particular interest to learn the dynamic behavior of mechatronic applications [15]. Ensembles of interconnected RNNs proved their usefulness in the modeling of complex dynamical systems [16]. However, as observed in [17], there is no physical interpretation given to this internal state which makes the convergence highly relying on hyperparameter tuning and initialization choices.

Having physical interpretations to the modeling can however be useful, especially in the context of servo systems, to distill a state-space representation. The dynamic behavior of the
system states (e.g. linear/angular positions and speeds) are mostly comprehended by the so-called derivative function in a partial and/or ordinary differential equations based model. Feedforward neural networks have been proposed in physics-informed deep learning [18] to approximate derivative functions of dynamical systems that unlike traditional black-box approximations return more interpretable models. Other recent approaches based on symbolic regression [19] and sparse regression [20], [21] were able to return more interpretable models by discovering the underlying governing equations.

Combining physics-based and data-driven methods and their respective advantages, has been a natural way to approach actual systems through models. These so-called grey-box models traditionally rely on either complementing a white-box model with a black-box model, or vice versa. Physical laws can be included in the loss function of black-box neural networks to guide the training process towards physical consistent model predictions [22]. Conversely, neural network mappings can compensate for the prediction discrepancies of a full physics-based model [23], [24]. These neural networks serve as nonlinear mappings that transform the state estimations in a sequential manner. They allow to improve predictions without representing any interpretable physical phenomena. In [25], unknown physical phenomena were learned by training a neural network using supervised learning that was subsequently inserted within a state-space derivative function.

This paper proposes a neural network augmented physics (NNAP) model, encompassing a neural network that is closely merged within a (physics-based) state-space derivative function, for servo systems exhibiting partially unknown dynamics with specific application to servo system consisting of bar linkage mechanism. The neural network complements for the unknown nonlinear dynamics and is simultaneously updated with the identification of the physical parameters. The overall aim of the presented methodology is to provide increased multistep prediction capabilities. These NNAP models additionally provide a means to explain and interpret certain phenomena that are otherwise difficult to physically model ab-initio. The effectiveness of the methodology is extensively applied and demonstrated on an experimental slider-crank system consisting of a servo drive and mechanical links with unknown nonlinear load force interactions. The paper is organized as follows. In Section II the slider-crank system with physical model and practical setup are presented. Section III presents the NNAP modeling formalism and the corresponding implementation details. Results of the predictive NNAP models are presented in Section IV. The identification of the key physical parameters and the forces acting on the slider-crank are presented together with the interpretability of the NNAP models.

II. SLIDER-CRANK SYSTEM

Various industrial applications require reciprocating linear motions. Motion control of these systems can be provided by means of linear electric motors [26]. For high power applications a combination of rotary motors in combination with mechanical transmissions [27] such as gears, cam-follower and bar linkage systems are the preferred driveline as they achieve energy efficient and robust operation. These drivelines exhibit highly nonlinear behavior that is often challenging to capture using first principle physics modeling. They are often plagued by unidentified load disturbances and unknown interactions of the system with the environment that are beyond the expert knowledge. These nonlinear phenomena are henceforth formalized by relation \( P \) that relates certain inputs of the system (e.g. velocity) to a response of the system (e.g. friction force). Next to these nonlinear unknown phenomena physical parameters such as the inertia in the servo drive system can be uncertain and are formalized as \( p \).

This paper applies and aims at validating a methodology to accommodate for unknown \( P \) and \( p \) that can arise in servo controlled systems. We consider a slider-crank mechanism translating a rotary motion into a linear displacement that is subject to an unidentified load. This application is of direct relevance in various industrial applications such as compressors [28], hydraulic pumps [29], weaving looms [30] and presses [31]. A system model comprehending the dynamics of the slider-crank system is built and experimental data is collected from a practical setup.

A. System model

The slider-crank system considered in this paper is schematically depicted in Fig. 1. The setup consists of a rotary servo motor delivering torque \( T \) to a first mechanical link that rotates with angle \( \theta \), that on its turn is connected to a second link. The latter link is connected to a slider (with relative angle \( \phi \) between this link and slider). The dynamics of the considered slider-crank setup are governed by force interactions between the rigid mechanical links. Appendix A details the expressions that formalize the dynamics of the multibody system. A time invariant state-space model can be distilled that comprehends the dynamics of the state \( x = [\theta \quad \omega]^T \). The angular speed is denoted as \( \omega = \dot{\theta} \). Note that the slider position \( d \) is geometrically coupled to motor angle \( \theta \) and angle \( \phi \) that on its turn is related to \( \theta \) via \( \phi = \arcsin \left( \frac{d}{l_2} \sin(\theta) \right) \). The nonlinear state space relation can be comprehended by a derivative function that relates the state \( x \) and control input \( u = T \) to the next state. We furthermore parameterize the state space model with physical parameters \( p \), being masses, lengths, inertias, rotational damping coefficient, relative lengths, that appear in (8). Finally, the load force \( F \) depicted in Fig. 1 is the unknown nonlinear relation \( P \) that is an additional disturbance input of the model. Given the above, we formalize the dynamics in
the slider-crank thus as $\dot{x} = f(x, u, F; p)$ with a geometrical relation between the linear displacement and velocity

$$\gamma(x; p) = \begin{bmatrix} d \\ \rho \end{bmatrix} = \begin{bmatrix} l_1 \cos(\theta) + l_2 \cos(\phi) - (l_2 - l_1) \\ -l_1 \sin(\theta) \omega - l_2 \sin(\phi) \dot{\phi} \end{bmatrix}$$

(1)

that is again dependent on the physical parameters $p$.

B. Practical setup

The practical slider-crank setup is depicted in Fig. 2. A 3 kW brushless servo motor with integrated drive is connected to a rigid crank having length $l_1 = 0.05$ m. The slider mechanism is connected to the crank via a rigid connecting rod with approximate length $l_2 = 0.29$ m. The rotary motion of the motor is measured by an incremental encoder (8192 CPR). An additional linear incremental encoder (grating pitch 20 $\mu$m) is used to measure the linear motion of the slider. A compression spring is added to invoke additional nonlinear disturbances in the external force $F$ acting on the slider. The spring force characteristic as function of displacement $d$ is given in Fig. 3. This characteristic was empirically determined for validation purposes later on.

Fig. 2: Slider-crank setup with spring load indicated by red circle.

Experimental data is collected from the slider-crank setup for different torque signal inputs. These torque profiles, see Fig. 4, are chosen ad-hoc to have diverse excitations of the system. These torque inputs are each time applied starting from two distinct motor positions leading to a total of 20 different trajectories of rotational ($\theta$ and $\omega$) and linear ($d$ and $v$) movement, as shown in Fig. 5. The measurements are sampled at 2000 Hz, resulting in 800 samples per trajectory.

Fig. 3: Spring characteristic.  Fig. 4: Torque signals.

III. NEURAL NETWORK AUGMENTED PHYSICS MODEL

Having accurate predictive models of systems such as the slider-crank setup presented in Section II can advance the design and motion control of various manufacturing and industrial systems. To accommodate for partially unknown dynamics (unknown nonlinear disturbance $P$ and uncertain physical parameters $p$) arising in servo systems, the physics-based ordinary differential equations (ODE) are complemented with additional data-driven models. The physics-based derivative function $f$ defines the behavior of the system state $x$ for given control input $u$. The known physical laws are encapsulated within $f$ and characterized by physical parameters $p$. The lack of knowledge about the unknown phenomena $P$ is compensated by introducing an additional input $z$.

$$x = f(x, u, z; p)$$

(2)

The neural network augmented physics (NNAP) model here includes an artificial neural network $\eta$ with the weights and biases of the neural nodes as learnable parameters $\alpha$ to compensate for these unknown phenomena $P$. The state measurements $x$ and control input $u$ are used to simultaneously update the neural network variables $\alpha$ and physical parameters $p$ to match the actual physical dynamical system behavior. Note that here contrary to [25] the data-driven model is not learned separately based on additional measurement data of the unknown phenomena $P$. A feedforward and a recurrent physics-based neural network model formalism are presented in the two next subsections respectively. Subsequently, the optimization process is explained, followed by the implementation details on the slider-crank mechanism.

A. Feedforward NNAP model

A feedforward model predicts the next state $x_{k+1}$ for given control input $u_k$ and current state $x_k$. By repeatedly feeding the predicted state back as input during the subsequent time instance the model becomes suitable for multistep predictions.
The universal approximation theorem states that feedforward neural networks with one hidden layer can approximate any continuous function for inputs within a specific range [32]. More specifically, we define $\eta$, a one hidden layer Rectified Linear Unit (ReLU) network of $n_h$ hidden units predicting a $n_o$ dimensional output. A ReLU model is chosen due to its proven usefulness in various regression tasks [11], [33]. The $n_i$ dimensional input $q \in \mathbb{R}^{n_i}$ to the neural network is a physics inspired static (nonlinear) mapping $g(x, u; p)$ of system state and control input information. The input vector is scaled by the standard deviation ($\sigma$) and control input information. The input vector is scaled by the corresponding transformation matrix $\Sigma = \text{diag} \{\sigma_1^{-1}, \ldots, \sigma_n^{-1}\}$ that needs to be tuned depending on the complexity of the problem. Here we assume the dimensions of the state of system state $\{h_j, j = 1, \ldots, n_i\}$ of the hidden layer. The hidden unit activation function $\Upsilon$ will, after adding the bias vector $b_h \in \mathbb{R}^{n_h}$, include the required nonlinearity in the function $\eta$. Subsequently, the output is obtained via a linear output function $\eta \Upsilon(\cdot) = \max(0, \cdot)$.

The architecture of the feedforward NNAP models is given in Fig. 6. The output of the ReLU network $\eta$, initialized by random weights and biases included in $\alpha$, replaces the variable $z$ in the ODE layer (2). Consequently, the assumption is that $q$ contains sufficient input information to estimate $z$, by approximating $P$ via a neural network $\eta$. We distill the following derivative function at each time instant $k$.

$$\dot{x}_k = f(x_k, u_k, \eta(q_k; \alpha); p)$$

(4)

The derivative function has now a hybrid nature and is further used to propagate to the next state via Euler's method by choosing $\Delta t$ equal to the fixed sampling time of the measurement data. The combination of interconnected layers boils down to an overall feedforward network function $M(x_k, u_k; p, \alpha)$ that estimates the state $x_{k+1}$ given the prior state $x_k$ and input $u_k$.

$$x_{k+1} = M(x_k, u_k; p, \alpha)$$

(5)

The feedforward NNAP model $M$ predicts only the state of the subsequent timestep. The model $M$ encompasses a static relation that can describe dynamical behavior once prior state predictions are fed back as model input.

B. Recurrent NNAP model

Recurrent neural networks (RNN) can cope with timeseries due to their ingrown dynamic nature [11], [34]. These models are penalized for estimation errors over a larger time horizon, aiming for improved multistep prediction capabilities. Fig. 7 illustrates a schematic overview of a basic RNN. The operator $D$ induces a one-step time delay of the internal state $h$. If we unfold the RNN in time we obtain a computational graph that contains shared weights $\psi$ at each time instance $j$. The total prediction error is an accumulation of the error at each time step. The gradients of this virtual multilayer network are required to update $\psi$ and can be derived via backpropagation through time (BPTT). BPTT calculates the gradients based on the chain rule of differentiation [34]. A RNN model accepts an input sequence of $N$ steps by processing each input element $r_j$ for $j \in \{1, \ldots, N\}$ at a time. These models contain information about prior inputs by updating an internal state vector $h_j$ via a (nonlinear) function $\mathcal{W}_h$. Thereafter, an output relation $\mathcal{W}_y$ maps the internal state to the model output $y$. The functions $\mathcal{W}_h$ and $\mathcal{W}_y$ typically contain network parameters $\psi$ that are optimized via gradient based algorithms.

$$h_j = \mathcal{W}_h(r_j, h_{j-1}; \psi)$$

$$y_j = \mathcal{W}_y(h_j; \psi)$$

(6)

A recurrent NNAP model $\mathcal{R}$ is constructed to predict a state trajectory $\{x_{k+1}, \ldots, x_{k+N}\}$ for given input sequence $\{u_k, \ldots, u_{k+N-1}\}$, starting from state $x_k$. The dimension of the internal state vector $h$ is typically a hyperparameter that needs to be tuned depending on the complexity of the problem. Here we assume the dimensions of the state of $h$ equal to the dimension of the state vector $x$. The corresponding initial value $h_0$, typically initialized as zero, is chosen equal to $x_k$. This allows to use the feedforward NNAP function $M$, detailed in Fig. 6, as update function $\mathcal{W}_h$ of the internal state $h$ in (6). This forces the hidden state $h$ to mimic the behavior of the system state $x$. Consequently, the output function $\mathcal{W}_y$ in (6) includes an identity transformation since we can consider $y_j = h_j = x_{k+j}$ for $j \in \{1, \ldots, N\}$. Fig. 8 illustrates the architecture of $\mathcal{R}$. The identification of the recurrent NNAP model $\mathcal{R}$ parameters $\psi$ boils down to optimizing the variables $p$ and $\alpha$ included in the feedforward NNAP model $M$ in (5).

$$[x_{k+1}, \ldots, x_{k+N}] = \mathcal{R}(x_k, u_k, \ldots, u_{k+N-1}; p, \alpha)$$

(7)
C. Optimization

The measurements $s$ of the system state $x$ are used to optimize the parameters $p$ and $\alpha$ of the NNAP model by minimizing the mean squared prediction error (MSE). The parameters $p$ and $\alpha$ are updated in the direction of the gradient of the MSE with respect to these parameters. The training data, containing the state trajectories $s$ of the MSE with respect to these parameters. The training series. Nevertheless, once the recurrent model $R$ latter ingrains BPTT to allow robust predictions of longer time

can be reworked to a more simple feedforward structure $M$ difference between the feedforward model $M$ for both the ReLU network and customized physics layers. The NNAP model by $\alpha$ includes identifying both the neural network parameters $\alpha$ and the set of physical parameters $\{J_1, B_m, m_3\}$ as subset of $p$ as will be further discussed in Section IV-B. Unless otherwise stated, we will perform the experiments on the feedforward architecture $M$ since inherently both architectures $M$ and $R$ include the same parameters to be identified. Leave-one-out-cross-validation (LOOCV) is used to assess the performances of the algorithm. This implies that the model is trained by 19 signals and validated by the trajectory excluded from the training set. This is repeated for all 20 signals to have a fair benchmark of the performances.

IV. RESULTS AND DISCUSSION

A. Dependency of multistep prediction accuracy to inputs

The ReLU network $\eta$ is assumed to approximate the unknown relation $P$ within the servo system once $M$ converges. However, since the physical load function $P$ is not known, the required input $q$ needs to be determined via experiments. Fig. 10b illustrates the time evolution of the angular velocity $\omega$ on the basis of the discrete model $M$ for test input $u = T_0$ and various inputs $q$, with $q_i \in \{\theta, \omega, d, v, T\}$, to the neural network $\eta$. We perform a multistep prediction as the state $x_{k+1}$ in (5) is the subsequent state input $x_k$ to $M$ in the next time instant. Errors are propagated each time when $M$ is evaluated, explaining the diverging trajectories with respect to the reference signal. The multistep predictive capabilities are benchmarked by calculating the root mean squared error (RMSE) of the predictive versus the reference (actual) trajectory $\omega$. Results are summarized in Fig. 10a by 80% confidence intervals having the colors corresponding to the inputs $q$ defined in Fig. 10b. Note that the measured data only contains $\theta, \omega$ and $T$. Inputs $d$ and $v$ originate from the internal mapping $g(x; u; p)$ detailed in (1). The lowest average RMSE is obtained for $q = [d, v]^T$. This gives a strong indication that the unknown function $P$, referring to the force $F$, is dependent on the position $d$ and speed $v$ of the slider. This insight makes sense knowing that the spring load and possible additional friction phenomena, both contained in the overall force $F$, are not included in the physics equations and thus should be absorbed by the network $\eta$. The set of experiments for which the required input information (i.e. set $\{d, v\}$ or the equivalently $\{\theta, \omega\}$) is not fed to the neural network $\eta$ clearly lead to inferior trajectory predictions. This experiment revealed the importance of the input information to the neural network $\eta$, at the same time showing that the NNAP model $M$ converges if indeed sufficient information is fed to $\eta$.

D. Implementation

Neural network augmented physics models are constructed for the slider-crank mechanism detailed in Section II. The unknown load force (unknown phenomena $P$) is accommodated by a neural network, i.e. $z \equiv F$. The ReLU neural network $\eta$ contains one hidden layer of $N = 32$ hidden units, determined via hyperparameter tuning, followed by a linear output layer of one unit. The input mapping $g(x; u; p)$ defining the ReLU input $q$ includes the identity transformation for $q_i \in \{\theta, \omega, T\}$ and the geometrical relations $\gamma(x; p)$ in (1) for $q_i \in \{d, v\}$. The experimental data from Fig. 4, 5a and 5b is used to train the NNAP model structures $M$ and $R$. The training data is fed into the neural network in mini-batches of 200 samples. The loss function is defined as the mean squared error (MSE) of the angular velocity $\omega$ predictions at each timestep. The NNAP model is optimized by an Adam optimizer [37] with learning rate of 0.0001. The optimization of $M$ (or $R$) includes identifying both the neural network parameters $\alpha$ and the set of physical parameters $\{J_1, B_m, m_3\}$ as subset of $p$ as will be further discussed in Section IV-B. Unless otherwise stated, we will perform the experiments on the feedforward architecture $M$ since inherently both architectures $M$ and $R$ include the same parameters to be identified. Leave-one-out-cross-validation (LOOCV) is used to assess the performances of the algorithm. This implies that the model is trained by 19 signals and validated by the trajectory excluded from the training set. This is repeated for all 20 signals to have a fair benchmark of the performances.

Fig. 8: Recurrent NNAP modeling formalism.

Fig. 9: Schematic overview of training data specifications.
Note that once the physical parameters have converged, further model predictions on unseen data during the training process. A subset of 10% require global optimization methods. Fig. 11b depicts the incorporation of the entire set of physical parameters; however, the ad hoc physical parameters was here determined since the variables are chosen fixed (see Table I). The selection of these parameters have on the model behavior, illustrated in the sensitivity analysis in Appendix B. The remaining less sensitive physical variables are chosen fixed (see Table I). The selection of physical parameters was here determined ad hoc since the incorporation of the entire set of physical parameters would require global optimization methods. Fig. 11b depicts the decreasing loss during the training process. A subset of 10% of the training data is chosen as validation data to monitor the model predictions on unseen data during the training process. Note that once the physical parameters have converged, further improvements are barely obtained.

Fig. 10: Influence of the input information included in q. The accuracy of the multistep predictions of test signals via LOOCV are summarized in a 80% confidence interval. A prediction of test signal T₉ is illustrated on the right.

**B. Simultaneous NNAP model optimization**

Henceforth, we use \( q = \begin{bmatrix} d & v \end{bmatrix}^T \) as input to the ReLU network \( \eta \). The optimization of the NNAP model \( M \) implies a simultaneous gradient based optimization of both \( p \) and \( \alpha \). Fig. 11a illustrates the convergence history of three physical parameters \( p \) (related to the rotational inertia \( J \), the rotational damping coefficient \( B \), and the mass \( m \) of the slider) in the gradient based optimization process. In [38] we show on numerical simulation data that the obtained values after convergence are indeed the true system parameters. The reason for their convergence lies in the high influence these parameters have on the model behavior, illustrated in the sensitivity analysis in Appendix B. The remaining less sensitive physical variables are chosen fixed (see Table I). The selection of physical parameters was here determined ad hoc since the incorporation of the entire set of physical parameters would require global optimization methods. Fig. 11b depicts the decreasing loss during the training process. A subset of 10% of the training data is chosen as validation data to monitor the model predictions on unseen data during the training process. Note that once the physical parameters have converged, further improvements are barely obtained.

**C. Recurrent NNAP model multistep prediction**

The training process of a feedforward NNAP model \( M \) only penalizes the prediction error of the next time step. A recurrent NNAP model \( R \) on the other hand penalizes the average accuracy of a trajectory sample. Since the model learns from the general dynamics this way, multistep predictions can be potentially improved. The influence of the length of the trajectory sequence \( N \) of the recurrent model \( R \) of the slider-crank system is depicted in Fig. 12a. The aforementioned converged model \( M \) with \( q = \begin{bmatrix} d & v \end{bmatrix}^T \) is used to initialize \( R \). Each model \( R \) is trained for 300 epochs in order to check if these recurrent structures can improve the prediction capabilities. The accuracy of the feedforward model \( M \) after 300 additional training epochs serves as reference. The LOOCV reveals that the recurrent model \( R \) in general does not perform better for unseen trajectories for small values of \( N \). However, the accuracy on test data clearly outperforms the feedforward model \( M \) for training on larger time sequences \( N \). The average RMSE of the test data was reduced by 33.4% for a model with \( N = 300 \) compared to the feedforward model \( M \). Multistep predictions based on trajectories included in the training set were directly improved for all recurrent models \( R \). The recurrent model \( R \) with \( N = 300 \) timesteps led to an average RMSE reduction by 46.9% compared to \( M \) on training data. This improved prediction accuracy however comes with increased computational costs as recurrent neural networks virtually unfold into more complex multilayer networks in order to apply BPTT (as illustrated in Fig. 7). The required computational time (training time per epoch) almost follows a power-law relation w.r.t. the sequences length \( N \) used during the training, as can be observed in Fig. 12b. This experiment was performed on a processing unit including a 6-core Intel i5-8400 CPU with 16 GB RAM. Fig. 13a illustrates an example of a multistep prediction of 800 time instances by a NNAP model for a test data trajectory corresponding to test input \( T₉ \). The recurrent neural network \( R \) indicates better prediction performances compared to \( M \). The higher the length \( N \) of the time sequences used during training, the more the neural network augmented physics models is able to capture the global dynamics of the slider-crank mechanism. Deviations of the predictive model with respect to measurements remain then limited as shown in Fig. 13b.
D. Retrieving physical insights

Next to obtaining a more precise predictive model, the presented methodology provides additional insights on the partially unknown dynamics $P$ (load force $F$) acting on the mechatronic slider-crank mechanism. At initialization, the neural network has no physical meaning (no resemblance with the actual $P$) as the neural network is defined by random learnable parameter values $\alpha_0$. The extracted neural network having as inputs $d$ and $v$ and as output $z = F$ is depicted in Fig. 14a. In that phase of training, totally wrong estimations of $z$ are used as input to (2) resulting in high initial loss (Fig. 11b). After convergence to a predictive NNAP model $M$ (or $R$), the ReLU network $\eta$ with optimized parameter values $\alpha^\ast$ can be extracted, see Fig. 14b.

![Fig. 14: Extracted ReLU network $\eta$ for different model parameters $\alpha$. The black dots indicate the data which was passed through the network during training.](image)

The discrepancy between the discovered force relation $\eta$ and the spring force $F_c$ raises the interest to elaborate further on the interpretability of $\eta$. The overall horizontal force acting on the slider is therefore considered as a summation $F = F_c + F_{nc}$ of a conservative ($F_c$) and non-conservative ($F_{nc}$) force. A conservative force is characterized by the property that the work done by moving the object between two points is independent of the taken path. In practice, we consider a conservative force $F_c(d)$, being only dependent on the position $d$ of the slider. Furthermore, the dissipative force $F_{nc}(d, v)$ relies on both the position $d$ and linear speed $v$. Therefore we replaced the ReLU network $z = \eta(d, v)$ by a new relation $z = \eta_c(d) + \eta_{nc}(d, v)$ deploying a summation of two separate feedforward ReLU networks $\eta_c(d)$ and $\eta_{nc}(d, v)$ that are trained in parallel within the NNAP model $M$. The solution of $\eta_c(d)$ and $\eta_{nc}(d, v)$ is however not unique since $\eta_{nc}(d, v)$ includes the inputs of $\eta_c(d)$. In practice, an additional $L_2$ regularization term $(c \cdot \eta_{nc}^2)$, with regularization parameter $c$, was added to the loss function. Fig. 16 illustrates the identified network components for an experimentally tuned regularization parameter $c = 10^{-6}$. The obtained results are the average predictions of $\eta_c$ and $\eta_{nc}$ evaluated by the training data for 10 converged neural network augmented physics models $M$. The ReLU network $\eta_c$ is able to accurately predict the behavior of the conservative spring force $F_c$ as can be observed in Fig. 16a. The regularization was able to successfully address the spring force to the ReLU network $\eta_c$. The remaining unknown dynamics attributed to the $\eta_{nc}$ is shown in Fig. 16c. From the corresponding side view in Fig. 16e we are able to discover the friction pattern that is dependent on the direction of the slider.

To validate the methodology that was able to retrieve physical insights, the aforementioned experiment is repeated on the experimental setup having no compression spring. Hence the state dependent conservative force is removed. From Fig. 16b, it can be seen that the discovered conservative force model $\eta_c$ trends towards zero. The non-conservative force can be extracted from $\eta_{nc}$ and is depicted in Figs. 16b and 16f. The non-conservative forces exhibit similar behavior as those shown in 16c and 16e. These results suggest the ability of neural network augmented physics models to provide insights on the on-going conservative and non-conservative forces of a mechatronic system interacting with its environment.
Neural network augmented physics models were presented in this paper to comprehend the behavior of servo systems that contain partially unknown dynamics due to unknown load forces and uncertain physical parameter values. The proposed NNAP model closely merges neural network models with physics-based state-space derivative functions. Experimental validation of this methodology is performed on a slider-crank system consisting of a servo drive and mechanical links. The dynamics of this system are only partially known since the load interactions, consisting of a compression spring and friction phenomena, are \emph{a priori} not identified. The unknown phenomena are captured by a neural network model and coupled to the physics-based model where the physical parameters are simultaneously identified with the learnable parameters of the neural network based on experimental data from the slider-crank system. Via automatic differentiation the errors were backpropagated through the layers resulting in convergence of the neural network augmented physics models. Two NNAP modeling formalisms were presented: a feedforward and a recurrent formalism. Accurate prediction capabilities were demonstrated on the slider-crank system using both formalisms and the recurrent formalism was able to reduce the prediction error up to 33.4% on validation data. The neural network component within the converged NNAP model of the slider-crank was extracted and provided physical insights on the load force. We successfully validated the possibility of discovering the unknown spring characteristic that was acting on the slider-crank system as well as the friction phenomena. Further research can be devoted to incorporate more unknown physical phenomena using multiple neural networks connected to multiple physical models as well as researching the ability to improve the discovery of the unknown phenomena with more elaborate regularization strategies.

\section*{Appendix A}

\textbf{Physics-based model}

The physics-based dynamical model of the slider-crank mechanism is derived by considering the simplified multibody system in Fig. 17. The variable $J_1$ includes the rotational inertia of both motor and crank with respect to the rotation axis. The rotational inertia $J_2$ of the connection rod is defined within its center of mass. The absolute distance towards the center of mass is defined as $r_1 = c_g l$. Furthermore we define $r' = l - r$ and let the subscripts $r_x$ and $r_y$ indicate the projection of $r$ on respectively the $x$-axis and $y$-axis.

$$m_1 \ddot{x}_1 = F_{ax} + F_{bx}$$
$$m_1 \ddot{y}_1 = -m_1 g + F_{ay} + F_{by}$$
$$J_1 \ddot{\omega} = T - B_m \dot{\omega} - r_1 x m_1 g + l_1 x F_{by} - l_1 y F_{bx}$$
$$m_2 \ddot{x}_2 = -F_{bx} + F_{cx}$$
$$m_2 \ddot{y}_2 = -F_{by} + F_{cy} - m_2 g$$
$$J_2 \ddot{\phi} = -r'_{2x} F_{by} - r'_{2y} F_{bx} - r_{2x} F_{cy} - r_{2y} F_{cx}$$
$$0 = -F_{cx} - m_3 g + F_N$$

The aforementioned equations are combined together with the geometrical relations, illustrated in Fig. 1, towards a physics inspired dynamic model $\dot{x} = f(x, u, F)$ with state $x = [\theta \ \omega]^T$. The highly nonlinear expression of $f$ is derived via symbolic solvers and is due to its complexity not mentioned in this paper.

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{slider_crank.png}
\caption{Multibody dynamical scheme of slider-crank mechanism.}
\end{figure}
APPENDIX B
SENSITIVITY ANALYSIS

The identified physical parameters \( p \) of the slider-crank mechanism are depicted in Table I. As was mentioned in Section IV-B, the parameter set \( \{ J_1, B_{in}, m_3 \} \) is optimized during the identification process of the NNAID model. Fig. 18 illustrates the sensitivity of the model parameters using the root mean square error (RMSE). It indicates the significant influence these parameters have on the model behavior so that convergence could be obtained. The remaining physical variables could be measured experimentally or determined via the CAD drawings of the setup.

**TABLE I: Identified parameters of slider-crank setup.**

| Parameter | Value 1 | Value 2 |
|-----------|---------|---------|
| \( m_1 \)  | 0.23 kg |         |
| \( m_2 \)  | 0.348 kg|         |
| \( m_3 \)  | 0.753 kg|         |
| \( B_{in} \) | 0.0087 Nm | 0.66 |

**Fig. 18:** Sensitivity analysis of physical parameter \( p \).

**REFERENCES**

[1] S. Kim, “Moment of inertia and friction torque coefficient identification in a servo drive system,” IEEE Trans. Ind. Electron., vol. 66, no. 1, pp. 60–70, 2019.

[2] D. Papageorgiou, M. Blanke, H. H. Niemann, and J. H. Richter, “Robust backlash estimation for industrial drive-train systems-theory and validation,” IEEE Trans. Control Syst. Technol., 2018.

[3] X. Wang, W. Wang, L. Li, J. Shi, and B. Xie, “Adaptive control of dc motor servo system with application to vehicle active steering,” IEEE/ASME Trans. Mechatronics, 2019.

[4] H. Ziyuan, H. Bangcheng, and L. Yun, “An explicit nonlinear model predictive abs controller for electro-hydraulic braking systems,” IEEE Trans. Ind. Electron., 2019.

[5] A. Kamadan, G. Kiziltas, and V. Patoglu, “Co-design strategies for optimal variable stiffness actuation,” IEEE/ASME Trans. Mechatronics, vol. 22, no. 6, pp. 2768–2779, 2017.

[6] S. J. Kim, K.-S. Kim, and D. Kumi, “Feasibility assessment and design optimization of a clutchless multimode parallel hybrid electric powertrain,” IEEE/ASME Trans. Mechatronics, vol. 21, no. 2, pp. 774–786, 2015.

[7] T. B. Schön, A. Wills, and B. Ninness, “Identification of nonlinear state-space models,” Automatica, vol. 47, no. 1, pp. 39–49, 2011.

[8] J. Kocijan, Modelling and control of dynamic systems using Gaussian process models. Springer, 2016.

[9] T. B. Wills, A.and Schön, L. Ljung, and B. Ninness, “Identification of hammerstein–wiener models,” Automatica, vol. 49, no. 1, pp. 70–81, 2013.

[10] S. A. Billings, Nonlinear system identification: NARMAX methods in the time, frequency, and spatio-temporal domains. John Wiley & Sons, 2013.

[11] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” Nature, vol. 521, no. 7553, p. 436, 2015.

[12] D. Kim, J. Kwon, S. Han, Y.-L. Park, and S. Jo, “Deep full-body motion network for a soft wearable motion sensing suit,” IEEE/ASME Trans. Mechatronics, vol. 24, no. 1, pp. 56–66, 2019.

[13] O. Janssens, R. Van de Walle, M. Loccufier, and S. Van Hoecke, “Deep learning for infrared thermal image based machine health monitoring,” IEEE/ASME Trans. Mechatronics, vol. 23, no. 1, pp. 151–159, 2018.

[14] O. Lakhal, A. Melingui, and R. Merzouki, “Hybrid approach for modeling and solving of kinematics of a compact bionic handling assistant manipulator,” IEEE/ASME Trans. Mechatronics, vol. 21, no. 3, pp. 1320–1335, 2015.

[15] O. Ogunnolu, X. Gu, S. Jiang, and N. Gans, “Nonlinear systems identification using deep dynamic neural networks,” arXiv preprint arXiv:1610.01439, 2016.

[16] N. Mohajerin and S. L. Waslander, “Modular deep recurrent neural network: Application to quadrupods,” in IEEE Int. Conf. on Systems, Man, and Cybernetics (SMC 2014). IEEE, 2014, pp. 1374–1379.

[17] ——, “State initialization for recurrent neural network modeling of time-series data,” in Int. Joint Conf. on Neural Networks (IJCNN 2017). IEEE, 2017, pp. 2330–2337.

[18] M. Raisi, P. Perdikaris, and G. E. Karniadakis, “Multistep neural networks for data-driven discovery of nonlinear dynamical systems,” arXiv preprint arXiv:1801.01236, 2018.

[19] F. Schmidt and H. Lipson, “Distilling free-form natural laws from experimental data,” Science, vol. 324, no. 5923, pp. 81–85, 2009.

[20] S. L. Brunton, J. L. Proctor, and J. N. Kutz, “Discovering governing equations from data by sparse identification of nonlinear dynamical systems,” Proc. of the Nat. Academy of Sciences, p. 201517384, 2016.

[21] Khatiry Goharoodi, K. Dekemele, M. Dupré, L.and Loccufier, and G. Crevecouer, “Sparse identification of nonlinear duffing oscillator from measurement data,” IFAC-PapersOnLine, vol. 51, no. 33, pp. 162–167, 2018.

[22] A. Karpate, W. Watkins, J. Read, and V. Kumar, “Physics-guided neural networks (pgnn): An application in lake temperature modeling,” arXiv preprint arXiv:1710.11431, 2017.

[23] N. Mohajerin and S. L. Waslander, “Multistep prediction of dynamic systems with recurrent neural networks,” IEEE Trans. Neural Netw. Learn. Syst., 2019.

[24] A. Ajay, M. Bauza, J. Wu, N. Fazelii, J. B. Tenenbaum, A. Rodriguez, and L. P. Kaelbling, “Combining physical simulators and object-based networks for control,” arXiv preprint arXiv:1904.06580, 2019.

[25] A. Punjani and P. Abbeel, “Deep learning helicopter dynamics models,” in IEEE Int. Conf. on Robotics and Automation. IEEE, 2015, pp. 3225–3230.

[26] I. Boldea, L. N. Tutela, W. Xu, and M. Pucci, “Linear electric machines, drives, and maglevs: an overview,” IEEE Trans. Ind. Electron., vol. 65, no. 9, pp. 7504–7515, 2018.

[27] C. Sharma and K. Purohit, Theory of mechanisms and machines. PHI Learning, 2006.

[28] Z. Gao, R. S. Colby, L. Turner, and B. Leprettre, “Filter design for estimating parameters of induction motors with time-varying loads,” IEEE Trans. Ind. Electron., vol. 58, no. 5, pp. 1518–1529, 2011.

[29] M. Li, R. Foss, K. Stelson, J. Van de Ven, and E. J. Barth, “Design, dynamic modelling and experiment validation of a novel alternating flow variable displacement hydraulic pump,” IEEE/ASME Trans. Mechatronics, 2019.

[30] G. Eren, R.and Ozkan and M. Karahan, “Comparison of heald frame motion generated by rotary dobbey and crank & cam shedding motions,” Fibres and textiles in eastern Europe, vol. 13, no. 4, p. 78, 2005.

[31] E. Zheng and X. Zhou, “Modeling and simulation of flexible slider-crank mechanism with clearance for a closed high speed press system,” Mechanism and Machine theory, vol. 74, pp. 10–30, 2014.

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

[33] P. Ramachandran, B. Zoph, and Q. V. Le, “Searching for activation functions,” arXiv preprint arXiv:1710.05941, 2017.

[34] R. Pascanu, T. Mikolov, and Y. Bengio, “On the difficulty of training recurrent neural networks,” in Int. Conf. on Machine Learning, 2013, pp. 1310–1318.

[35] F. Chollet et al., “Keras,” https://keras.io, 2015.

[36] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard et al., “Tensorflow: A system for large-scale machine learning,” in 12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16), 2016, pp. 265–283.

[37] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” arXiv preprint arXiv:1412.6980, 2014.

[38] W. De Groote, E. Kikken, S. Goyal, S. Van Hoecke, E. Hostens, D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” arXiv preprint arXiv:1412.6980, 2014.

[39] W. De Groote, E. Kikken, S. Goyal, S. Van Hoecke, E. Hostens, and G. Crevecouer, “Hybrid derivative functions for identification of unknown loads and physical parameters with application on slider-crank mechanism,” in 2019 IEEE/ASME Int. Conf. on Advanced Intelligent Mechatronics (AIM). IEEE, 2019.