Physics–Guided Neural Networks for Feedforward Control: From Consistent Identification to Feedforward Controller Design*

Max Bolderman¹, Mircea Lazar³, and Hans Butler¹,²

Abstract—Model–based feedforward control improves tracking performance of motion systems if the model describing the inverse dynamics is of sufficient accuracy. Model sets, such as neural networks (NNs) and physics–guided neural networks (PGNNs) are typically used as flexible parametrizations that enable accurate identification of the inverse system dynamics. Currently, these (PG)NNs are used to identify the inverse dynamics directly. However, direct identification of the inverse dynamics is sensitive to noise that is present in the training data, and thereby results in biased parameter estimates which limit the achievable tracking performance. In order to push performance further, it is therefore crucial to account for noise when performing the identification. To address this problem, this paper proposes a forward system identification using (PG)NNs from noisy data. Afterwards, two methods are proposed for inverting PGNNs to design a feedforward controller. The developed methodology is validated on a real–life industrial linear motor, where it showed significant improvements in tracking performance with respect to the direct inverse identification.

I. INTRODUCTION

Model–based feedforward control significantly improves tracking performance of motion systems if the available model describing the inverse system dynamics is of sufficient accuracy [1]. Typically, physics–based models, i.e., model sets derived from physical knowledge of the system, are used for feedforward control, see, e.g., [2]–[4]. However, the limited flexibility of physics–based models results in structural model errors when such models are used to describe the complete dynamics. Certainly, this becomes apparent when considering parasitic effects, such as electromagnetic distortions that arise from manufacturing tolerances [5].

In order to deal with parasitic effects or other nonlinear phenomena that are hard to model, neural networks (NNs) are proposed as models for identification of the inverse system dynamics in [6], see also [7], [8]. The universal approximation capabilities of NNs theoretically enable a perfect description of the inverse system if the NN dimensions are chosen sufficiently large [9]. Additionally, it was shown in [10], [11] that augmenting a physics–based feedforward with NNs improves tracking performance and robustness to non–training data.

Since NN models are not analytically invertible in general, inverse model–based feedforward control design using NNs or PGNNs, hinges on correctly identifying the inverse dynamics directly, also referred to as general learning in [6], [12]. However, using a direct inverse identification results in biased parameter estimates when noise is present in the data [13]. Consequently, if noise is present, the bias decreases the model accuracy, and thereby limits the achievable tracking performance of the corresponding feedforward controller [1]. An alternative is to adopt a so–called specialized learning approach [14], in which a NN is connected in series with the system to be identified. In this approach, the training procedure minimizes the difference between the system output and the NN input. However, specialized learning requires more complex operations when used either offline or online, see, e.g., [6] and [14] for details.

In this paper we develop a new solution to nonlinear model–based feedforward control design from noisy data by means of consistent identification of the forward dynamics, and online dynamic inversion of the forward model. First, we discuss the required assumptions to obtain a consistent estimate of the forward dynamics using PGNN model parametrizations for relevant noise structures. Then, we propose two methods for inverting the identified forward dynamics parametrized using (PG)NNs: a gradient–based numerical method that is inspired by techniques that are discussed in [15], and an analytic method suitable for a class of electromagnetic actuators common in high–precision motion control. Thereby, we no longer use a (PG)NN to directly parametrize the inverse dynamics, which is a prerequisite for both general and specialized learning using (PG)NNs. An extra benefit is that identification of the forward dynamics, i.e., by minimizing the difference between the measured output and the predicted output, is in line with the evaluation of the tracking performance achieved by the feedforward controller. Specialized learning also inherits this benefit, and it was also exploited in [16] by filtering the cost function with the process sensitivity in a direct inverse identification, i.e., by minimizing the difference between the actual and predicted input.

The main contributions of this paper are as follows. First, based on fundamental approaches discussed in [17] for system identification, we show that forward system identification using a PGNN model class results in parameter estimates that are consistent, i.e., unbiased with probability 1 when the data length goes to infinity. It is shown that the estimates remain consistent when the data is generated from a closed–loop experiment. Secondly, we derive methods for inversion of the identified PGNN describing the forward dynamics. Initially, a gradient–based technique is suggested.

*This work is part of the research programme 9654 with project number 17973, which is (partly) financed by the Dutch Research Council (NWO).
¹Control Systems Group, department of Electrical Engineering, Eindhoven University of Technology, The Netherlands. {m.bolderman, m.lazar}@tue.nl
²ASML, Veldhoven, The Netherlands. hans.butler@asml.nl
to find the feedforward input. Afterwards, an analytically invertible PGNN model is proposed for the case when the gradient–based technique is not implementable in real–time.

II. PRELIMINARIES AND PROBLEM STATEMENT
A. System dynamics and feedforward control
Consider the discrete–time, single–input single–output (SISO), nonlinear time–invariant system with autoregressive exogeneous (ARX) noise structure, such that

\begin{align}
y(t) = h(\phi(t)) + v(t), \\
\phi(t) = [y(t-1), ..., y(t-n_a)], \\
u(t-n_k-1), ..., u(t-n_k-n_b)]^T.
\end{align}

In (1), \( y(t) \) is the system output at time index \( t \), \( \phi(t) \) the regressor, \( u(t) \) the input, \( n_a \) and \( n_b \) describe the order of the dynamics, and \( n_k \) is the number of pure input delays. The function \( h : \mathbb{R}^{n_a+n_b} \rightarrow \mathbb{R} \) describes the system dynamics, and \( v(t) \) is assumed to be a zero mean white noise with variance \( \sigma^2_v := \lim_{N \to \infty} \frac{1}{N} \sum_{t=0}^{N-1} v(t)^2 \). The control input is obtained as

\begin{align}
u(t) = u_{ff}(t) + u_{fb}(t),
\end{align}

where \( u_{ff}(t) \) is the feedforward which enables high tracking performance, and \( u_{fb}(t) \) is a closed–loop stabilizing feedback input \([18]\). In the remainder of this work, we assume that a closed–loop stabilizing feedback controller is given.

Remark 2.1: The nonlinear ARX (NARX) structure (1) is mostly popular for its simplicity, and does often not describe noise as experienced in real–life, see Fig. 1. Although, we focus on NARX in this paper, we also highlight the output–error (NOE) and input–error (NIE) noise structures due to their direct relation with sensor and actuator noise, respectively. Other noise structures can be considered similarly.

From Fig. 1 we obtain the NOE dynamics

\begin{align}
y(t) = h(\phi_{NOE}(t)) + v(t), \\
\phi_{NOE}(t) = [y(t-1) – v(t-1), ..., y(t-n_a) – v(t-n_a), u(t-n_k-1), ..., u(t-n_k-n_b)]^T.
\end{align}

Similarly, the NIE dynamics is given as

\begin{align}
y(t) = h(\phi_{NIE}(t)), \\
\phi_{NIE}(t) = [y(t-1), ..., y(t-n_a), u(t-n_k-1) + v(t-n_k-n_b), ..., u(t-n_k-n_b) + v(t-n_k-n_b)]^T.
\end{align}

The feedforward input \( u_{ff}(t) \) is the input \( u(t) \) that yields \( y(t) = r(t) \) for system (1) with \( r(t) \) the reference, when \( v(t) = 0 \). Substitution of \( y(t) = r(t) \) and \( u(t) = u_{ff}(t) \) in (1), and shifting both sides \( n_k + 1 \) samples forward, gives

\begin{align}
r(t + n_k + 1) = h([r(t + n_k), ..., r(t + n_k – n_a + 1), u_{ff}(t), ..., u_{ff}(t – n_b + 1)]^T). 
\end{align}

Then, with a slight abuse of notation, let \( h^{-1} \) be the mapping describing the inverse dynamics, such that

\begin{align}
u_{ff}(t) = h^{-1}([r(t + n_k + 1), ..., r(t + n_k – n_a + 1), u_{ff}(t – 1), ..., u_{ff}(t – n_b + 1)]^T).
\end{align}

Unfortunately, the actual function \( h \) is unknown and, therefore, we cannot use \( h^{-1} \) to design the feedforward controller.

In the remainder of this paper, we use \( \phi(t) \) as the regressor for the forward dynamics, and \( \phi'(t) \) as the regressor for the inverse dynamics, e.g., \( \phi'(t) = [y(t + n_k + 1), ..., y(t + n_k – n_a + 1), u(t – 1), ..., u(t – n_b + 1)]^T \) for NARX. Similarly, \( \phi_{ff}(t) \) and \( \phi'_{ff}(t) \) are obtained by substitution of \( y(t) = r(t), u(t) = u_{ff}(t) \), and \( v(t) = 0 \) in \( \phi(t) \) and \( \phi'(t) \), respectively.

B. System identification procedure
Typically, a physics–based model parametrization is derived from first principle modelling.

Definition 1: A physics–based model is defined as

\begin{align}
y_{ff}(\theta_{phy}, \phi(t)) = f_{phy}(\theta_{phy}, \phi(t)),
\end{align}

where \( y_{ff}(\theta_{phy}, \phi(t)) \) indicates the prediction of the output \( y(t) \), and \( \theta_{phys} \) are the parameters of the physical model.

The parameters \( \theta_{phy} \) are chosen according to an identification criterion, such as the mean–squared error (MSE).

Definition 2: The MSE identification criterion is given as

\begin{align}
\theta_{phy}^* = \arg \min_{\theta_{phy}} \frac{1}{N} \sum_{t=0}^{N-1} (y(t) – y_{ff}(\theta_{phy}, \phi(t)))^2,
\end{align}

where the summation is taken over a data set \( Z^N = \{\phi(0), y(0), ..., \phi(N – 1), y(N – 1)\} \).

Following the same reasoning as for (6), after identification of the parameters \( \theta_{phy} \), the physics–based feedforward controller is given as

\begin{align}
u_{ff}(t) = f_{phy}^{-1}(\theta_{phy}^*, \phi_{ff}(t)),
\end{align}

where \( f_{phy}^{-1} \) indicates the inverse of \( f_{phy} \), which is assumed to be known (typically a physics–based analytic formula is used, e.g., inverse linear motion dynamics). In order to obtain an implementable physics–based feedforward as in (9), we assume knowledge of the reference up until time \( t + n_k + 1 \) at time \( t \), and assume that \( f_{phy}^{-1} \) is bounded–input bounded–output (BIBO) stable. Methods for obtaining a stable inverse of linear systems with unstable inverses are listed in \([19]\).

The physics–based model (7) does not capture the complete dynamics \( h \) in (1), due to the presence of parasitic effects, such as electromagnetic distortions \([5]\).

Definition 3: The unmodelled dynamics are \( g(\phi(t)) := h(\phi(t)) – f_{phy}(\theta_{phy}^*, \phi(t)) \), such that rewriting (1) gives

\begin{align}
y(t) = f_{phy}(\theta_{phy}^*, \phi(t)) + g(\phi(t)) + v(t).
\end{align}
As suggested in [10], we augment the physics–based model with a NN to identify also the unmodelled dynamics.

Definition 4: A PGNN is defined as
\[
y(\theta, \phi(t)) = f_{\text{phy}}(\theta_{\text{phy}}, \phi(t)) + f_{\text{NN}}(\theta_{\text{NN}}, \phi(t)),
\]
with \( \theta := \{\theta_{\text{phy}}, \theta_{\text{NN}}\} \) the PGNN parameters, and \( \theta_{\text{NN}} := \{W_1, B_1, ..., W_{l+1}, B_{l+1}\} \) the NN weights and biases with \( l \) the number of hidden layers. The NN output is
\[
f_{\text{NN}}(\theta_{\text{NN}}, \phi(t)) = W_{l+1} \alpha_1(\ldots \alpha_1(W_1 \phi(t) + B_1)) + B_{l+1},
\]
where \( \alpha_i \) denotes the aggregation of activation functions of layer \( i = 1, ..., l \).

Remark 2.2: The PGNN becomes a standard, black–box NN when no physical knowledge of the system is present, i.e., for \( f_{\text{phy}} = 0 \) in (11). Therefore, the methodology developed in this paper applies also to black–box NNs, with the exception of the analytic inversion in Section IV-B. In [11], it was demonstrated that using a PGNN improves training convergence and reduces sensitivity to the training data compared to using only a black–box NN.

The flexible nature of the NN can create an overparameterization in the PGNN (11) when training according to the MSE identification criterion (8), which results in a parameter drift during training. Therefore, a regularized MSE identification criterion was introduced in [11].

Definition 5: The regularized MSE identification criterion is given as
\[
\hat{\theta} = \arg\min_{\theta} \frac{1}{N} \sum_{t=0}^{N-1} (y(t) - \hat{y}(\theta, \phi(t)))^2 + \left(\theta_{\text{phy}}^* - \theta_{\text{phy}}\right)^T \Lambda \left(\theta_{\text{phy}}^* - \theta_{\text{phy}}\right),
\]
with \( \Lambda \) a positive definite matrix, and \( \theta_{\text{phy}}^* \) the solution of (8) for the physical model (7) contained within the PGNN (11).

The majority of literature on (PG)NN–based feedforward performs a direct inverse identification, see, e.g., [7], [10]. Essentially, we parametrize the inverse dynamics to predict
\[
\hat{u}(\theta, \phi'(t)) = f_{\text{phy}}^{-1}(\theta_{\text{phy}}, \phi'(t)) + f_{\text{NN}}(\theta_{\text{NN}}, \phi'(t)),
\]
and train \( \theta \) according to identification criterion
\[
\hat{\theta} = \arg\min_{\theta} \frac{1}{N} \sum_{t=0}^{N-1} (u(t) - \hat{u}(\theta, \phi'(t)))^2 + \left(\theta_{\text{phy}}^* - \theta_{\text{phy}}\right)^T \Lambda \left(\theta_{\text{phy}}^* - \theta_{\text{phy}}\right).
\]
If the data is noise free, this approach is more attractive for feedforward control design, since we directly obtain the inverse dynamics. However, when the data contains noise, parameter estimates become biased.

C. Problem statement

Since direct inverse system identifications result in biased parameter estimates in the presence of noisy data, nonlinear model–based feedforward controller design from noisy data remains an open problem. In this paper, we develop a systematic PGNN feedforward controller design procedure with the following characteristics:

1) Consistent parameter estimation: the PGNN consistently identifies the forward dynamics, including the unmodelled dynamics \( g(\phi(t)) \) in the presence of noise;
2) System inversion: in order to derive the PGNN feedforward, the identified forward dynamics must be either analytically or numerically invertible.

III. CONSISTENT PGNN IDENTIFICATION

Since the universal approximation theorem for NNs holds only within a compact domain [9], we define the operating conditions of the feedforward controller.

Definition 6: The operating conditions \( \mathcal{R} \) are defined as all possible regressors provided to the PGNN, such that
\[
\phi(t), \phi(t) \in \mathcal{R},
\]
for all \( t \), all references supplied to the PGNN feedforward, and all regressors in the data set \( Z^N \).

Then, following the framework in [17], it is possible to obtain consistent estimates of the system (1). Definition 7: A parameter estimate \( \hat{\theta} \) of \( \theta^* \) is consistent if \( \hat{\theta} \rightarrow \theta^* \) for \( N \rightarrow \infty \) with probability 1.

We adopt the following common assumptions on the model, data, and training to prove consistency for the PGNN.

Assumption 3.1: There exists a \( \theta_{\text{NN}}^* \) such that
\[
f_{\text{NN}}(\theta_{\text{NN}}^*, \phi(t)) = g(\phi(t)) \text{ for all } \phi(t) \in \mathcal{R}.
\]
Assumption 3.2: For \( \theta_{\text{NN}}^* \neq \theta_{\text{NN}}^B \) with \( f_{\text{NN}}(\theta_{\text{NN}}^A, \phi(t)) \neq f_{\text{NN}}(\theta_{\text{NN}}^B, \phi(t)) \) for some \( \phi(t) \in \mathcal{R} \), we have
\[
\frac{1}{N} \sum_{t=0}^{N-1} (f_{\text{NN}}(\theta_{\text{NN}}^A, \phi(t)) - f_{\text{NN}}(\theta_{\text{NN}}^B, \phi(t)))^2 > 0.
\]
Assumption 3.3: The optimization over \( \theta \) of the identification criterion (13) yields a global optimum.

Proposition 3.1: Consider the PGNN (11) that is used to identify the NARX system (1) according to identification criterion (13). Suppose that Assumptions 3.1, 3.2 and 3.3 hold. Then, for \( N \rightarrow \infty \), the identified PGNN parameters satisfy \( \hat{\theta} = \{\theta_{\text{phys}}, \theta_{\text{NN}}\} \rightarrow \{\theta_{\text{phys}}^*, \theta_{\text{NN}}^*\} \).

Proof: The proof follows the approach in [17], by showing that the globally minimizing argument of the cost function corresponds to a consistent estimate. Substitution of the system dynamics (10) and the PGNN (11) into the identification criterion (13) for \( N \rightarrow \infty \) gives
\[
\lim_{N \rightarrow \infty} \frac{1}{N} \sum_{t=0}^{N-1} \left( f_{\text{phy}}(\theta_{\text{phy}}^*, \phi(t)) - f_{\text{phy}}(\theta_{\text{phy}}, \phi(t)) \right)^2 + \frac{\sigma_v^2}{\sigma_e^2} \geq \sigma_v^2,
\]
where \( \sigma_v^2 \) occurs from \( v(t) \) which is taken outside of the MSE term, since it is uncorrelated with \( \phi(t) \). The MSE and regularization term are non–negative, such that the inequality in (18) holds with equality only if \( \theta_{\text{phy}} = \theta_{\text{phy}}^* \) (regularization term), and \( \theta_{\text{NN}} = \theta_{\text{NN}}^* \) (MSE term, substitute \( \theta_{\text{phy}} = \theta_{\text{phy}}^* \)).

It follows from Proposition 3.1 that \( f_{\text{phy}}(\theta_{\text{phy}}^*, \phi(t)) + f_{\text{NN}}(\theta_{\text{NN}}, \phi(t)) = h(\phi(t)) \) for all \( \phi(t) \in \mathcal{R} \). Therefore,
the identified PGNN replicates the system under the listed assumptions. Note that, even though Assumptions 3.1, 3.2, 3.3 may not hold in general, Proposition 3.1 offers an additional reliability for the forward identification compared to the inverse identification. Indeed, in the latter approach a bias is present due to the correlation between $\hat{y}(t)$ and $v(t)$, such that $v(t)$ cannot be taken out of the MSE term.

**Remark 3.1:** Consider the NOE structure (3), Fig. 1. Then, consistency is obtained by using the PGNN (11) with $\phi(t) = [\tilde{y}(\theta, \phi(t - 1)), ..., \tilde{y}(\theta, \phi(t - n_0))], v(t - n_k - 1), ..., v(t - n_k - m)],$ provided that the initial conditions, i.e., $v(t) - v(t)$ and $u(t)$ for $t < 0$, are known and $b$ is asymptotically stable. The proof follows similar to the proof of Proposition 3.1.

**Remark 3.2:** Consider the system with NIE noise (4), Fig. 1. Then, the responses caused by the input $u(t)$ and the noise $v(t)$ cannot be distinguished. In this situation, a direct inverse identification is beneficial, which is equivalent to NOE forward identification. Under the assumption that the inverse dynamics $h^{-1}$ in (6) exists and is stable, identification according to criterion (15) with inverse PGNN (14) and $\phi'(t) = [y(t + n_k + 1), ..., y(t + n_k - n_a + 1), \hat{\hat{\theta}}, \hat{\phi}'(t - 1)], ..., \hat{\hat{\theta}}, \hat{\phi}'(t - n_0 + 1)]^T$ yields consistent estimates.

**Remark 3.3:** Assume that the data $Z^N$ is generated under closed-loop operation, e.g., using a linear feedback

$$u(t) = C(q^{-1})(r(t) - y(t)) + \Delta u(t),$$

where $C(q^{-1})$ is the transfer function of the feedback controller, $q^{-1}$ the backwards shift operator, and $\Delta u(t)$ the excitation on the input during data generation. The feedback introduces a correlation between the input and output, and therefore, the noise as well. Still, the proof of Proposition 3.1 remains valid, because $\hat{\phi}(t)$ and $v(t)$ are uncorrelated (note that $u(t - n_k - 1)$ is the most recent input in $\phi(t)$).

From Remarks 3.1, 3.2, 3.3 it becomes apparent that, in order to obtain consistent estimates, the regressor $\phi(t)$ must be chosen to appropriately account for where the noise enters the system. There are several approaches that can be used to remove the bias when the regressor is not, or cannot be chosen appropriately. One example is the instrumental variable (IV) approach, which is for example used in [20]. Therein, it was shown that a bias-correction factor was required to obtain consistent estimates, which assumed specific knowledge of the noise distribution, variance, and structure. Recently, the IV approach was also applied to NN-based identification in [21].

### IV. Feedforward Controller Design

In the previous section we discussed the identification of the forward dynamics (1) using the PGNN (11). In this section, we use the identified PGNN, i.e., (11), with $\theta = \hat{\theta}$ from (13), to compute the feedforward control action, i.e., the input $u_{ff}(t)$ that should yield $y(t) = r(t)$ for all $t \geq 0$. Similar to (5), (6), this problem is defined as:

- compute $u_{ff}(t)$,
- such that $\hat{y}(\theta, \phi_{ff}(t + n_k + 1)) = r(t + n_k + 1).$ \hspace{1cm} (20)

Note that, (20) uses $t + n_k + 1$, because $u_{ff}(t)$ is the most recent input in $\phi_{ff}(t + n_k + 1)$.

#### A. Gradient-based numerical inversion

Let us define $V_{ff}(u_{ff}(t)) := r(t + n_k + 1) - \hat{y}(\theta, \phi_{ff}(t + n_k + 1)).$ Then, following (20), $u_{ff}(t)$ must satisfy

$$V_{ff}(u_{ff}(t)) := r(t + n_k + 1) - \hat{y}(\theta, \phi_{ff}(t + n_k + 1)) = 0. \hspace{1cm} (21)$$

We neglect the time indices for brevity, and compute the gradient

$$\frac{\partial V_{ff}(u_{ff})}{\partial u_{ff}} = -\frac{\partial f_{phy}(\hat{\theta}_{phy}, \phi_{ff})}{\partial u_{ff}} - \frac{\partial f_{NN}(\theta_{NN}, \phi_{ff})}{\partial u_{ff}},$$

(22)

where the first term is derived from the known physical model, and backpropagation gives

$$\frac{\partial f_{NN}(\hat{\theta}_{NN}, \phi_{ff})}{\partial u_{ff}} = W_{l+1} \frac{\partial \phi_{ff}}{\partial u_{ff}},$$

with $\beta_{i}(x_{i-1}) := \text{diag}(\frac{\partial \phi_{i}(x)}{\partial x}|_{x=x_{i-1}})$, and $x_{i-1}$ the output of layer $i - 1$ in the NN. Then, the gradient-based iterative search in Algorithm 1 is performed online for each time index $t$ to find the feedforward input $u_{ff}(t)$, i.e., solve (20).

**Algorithm 1** Search algorithm for feedforward $u_{ff}(t)$.

| Initialize $u_{ff}^{(0)}(t) = f_{phy}^{-1}(\theta_{phy}, \phi_{ff}(t)).$ |
| for $i \in \{1, ..., k\}$ do |
| Compute $V'_{ff}(u_{ff}^{(i-1)}(t)) = \frac{\partial V_{ff}(u_{ff}(t))}{\partial u_{ff}}|_{u_{ff}(t)=u_{ff}^{(i-1)}(t)}$ |
| Update $u_{ff}^{(i)}(t) = u_{ff}^{(i-1)}(t) - V_{ff}(u_{ff}^{(i-1)}(t)) V_{ff}^{(i)}(u_{ff}^{(i-1)}(t))$ |
| end for |
| Return $u_{ff}(t) = \arg \min_{i \in \{0, ..., k\}} \left| V_{ff}(u_{ff}^{(i)}(t)) \right|$ |

In Algorithm 1 the optimization starts from the physics-based feedforward, because it is often close to the optimal feedforward for the PGNN, see [11]. The updates are performed using the Newton–Raphson method, which generally converges in a limited number of iterations, but other optimization methods can also be used. It is desired to have the number of iterations $k$ large, to ensure that the solver converges. However, for real-time implementation, the number of updates is limited by the computation time, hardware, and sampling time of the system. In the last step of Algorithm 1, the best feedforward signal encountered is chosen as output, such that the physics-based feedforward is regained in the worst-case that the solver diverges. Actuator limitations can be accounted for by limiting the search within a specified domain, or by saturating $u_{ff}(t)$.

#### B. Analytical inversion

When $u(t - n_k - 1)$ does not pass through the NN part of the PGNN (14), we obtain a specific class of PGNNs which are analytically invertible, i.e.,

$$\hat{\phi}(\theta, \phi_{ff}(t + n_k + 1)) = f_{phy}(\theta_{phy}, \phi(t + n_k + 1)) + f_{NN}(\theta_{NN}, \phi_{ff}(t + n_k + 1)),$$

(24)
where \( \phi_y = [y(t-1), \ldots, y(t-n_a)]^T \) and \( \phi_u(t) = [u(t-n_k-2), \ldots, u(t-n_k-n_b)]^T \), such that \( \phi(t) = [\phi_y(t)^T, u(t-n_k)^T]^T \). Using (24) with \( \theta = \hat{\theta} \), we can directly compute the feedforward controller analytically

\[
\begin{align*}
  u_{\text{ff}}(t) &= f_{\text{phy}}^{-1}(\hat{\theta}_{\text{phy}}, \phi_y(t) - \Delta_f(t)), \quad \\
  \Delta_f(t) &= [1, 0, \ldots, 0]^T f_{\text{NN}}(\hat{\theta}_{\text{NN}}, r)(t+n_k), \ldots, \quad \\
    & \quad r(t+n_k-n_a+1), u_{\text{ff}}(t-1), \ldots, \quad \\
  & \quad u(t+n_k-n_b+1)]^T.
\end{align*}
\]

Since the invertible PGNN (24) is more restrictive compared to the PGNN (11), i.e., it does not pass \( u(t-n_k-1) \) through the NN, we show in the remainder of this section that, under some assumptions and a revision of the training cost function (13), the invertible PGNN (24) still yields consistent estimates. First, we choose \( \theta_{\text{phy}} = [\zeta^T, \psi^T]^T \), where \( \psi \) are the parameters that affect \( u(t-n_k-1) \), and \( \zeta \) are the remaining parameters. We assume that the physical model is able to capture the effect of the most recent input.

**Assumption 4.1:** There exists a \( \psi_0 \) such that the unmodelled dynamics \( g(\hat{o}(t)) \) does not depend on \( u(t-n_k-1) \).

Correspondingly, if we define \( \theta_{\text{phy}} = [\zeta^T, \psi^T]^T \), the dynamics (1) can be rewritten into

\[
y(t) = f_{\text{phy}}(\hat{\theta}_{\text{phy}}, \phi(t)) + g\left(\left[\begin{array}{c} \phi_y(t) \\ \phi_u(t) \end{array}\right]\right) + v(t).
\]

Since \( \psi^* \) resulting from the MSE identification (8) generally differs from \( \psi_0 \), it is omitted in the regularization, i.e.,

\[
\hat{\theta} = \arg \min_{\theta} \frac{1}{N} \sum_{t=0}^{N-1} (y(t) - \hat{y}(\theta, \phi(t)))^2 \quad + (\zeta^* - \zeta)\Lambda(\zeta^* - \zeta).
\]

Finally, an additional assumption is required on the data, since we no longer penalize \( \psi^* - \psi \) in the cost function.

**Assumption 4.2:** For some \( \psi^A \neq \psi^B \) with

\[
f_{\text{phy}}(\left[\begin{array}{c} \zeta^T, \psi^A \end{array}\right]^T, \phi(t)) \neq f_{\text{phy}}(\left[\begin{array}{c} \zeta^T, \psi^B \end{array}\right]^T, \phi(t))
\]

for some \( \phi(t) \in \mathcal{R} \), we have that

\[
\frac{1}{N} \sum_{t=0}^{N-1} \left( f_{\text{phy}}\left(\left[\begin{array}{c} \zeta^* \\ \psi^A \end{array}\right], \phi(t)\right) - f_{\text{phy}}\left(\left[\begin{array}{c} \zeta^* \\ \psi^B \end{array}\right], \phi(t)\right)\right)^2 > 0.
\]

**Proposition 4.1:** Consider the PGNN (24) that is used to identify the system (1) according to identification criterion (27). Suppose that Assumptions 3.1, 3.2, 3.3, 4.1, and 4.2 hold. Then, for \( N \to \infty \), the PGNN parameters are identified as \( \hat{\theta} = \{\hat{\theta}_{\text{phy}}, \hat{\theta}_{\text{NN}}\} = \{\theta_{\text{phy}}, \theta_{\text{NN}}^*\} \).

**Proof:** The proof follows similarly to the proof of Proposition 3.1, i.e., we substitute (26) and (24) into (27) and take \( v(t) \) out of the MSE term to obtain the cost function

\[
\begin{align*}
  \lim_{N \to \infty} \frac{1}{N} \sum_{t=0}^{N-1} \left( f_{\text{phy}}(\hat{\theta}_{\text{phy}}, \phi(t)) - f_{\text{phy}}(\hat{\theta}_{\text{phy}}, \phi(t)) \right) & + g\left(\left[\begin{array}{c} \phi_y(t) \\ \phi_u(t) \end{array}\right]\right) - f_{\text{NN}}(\hat{\theta}_{\text{NN}}, \left[\begin{array}{c} \phi_y(t) \\ \phi_u(t) \end{array}\right])^2 \\
  & + (\zeta^* - \zeta)^T \Lambda(\zeta^* - \zeta) + \sigma_v^2 \geq \sigma_v^2.
\end{align*}
\]

In (29) the equality holds only if \( \zeta = \zeta^* \) (regularization term), and \( \{\psi, \theta_{\text{NN}}\} = \{\psi_0, \theta_{\text{NN}}^*\} \) (MSE term, substitute \( \zeta = \zeta^* \) and observe that, when \( \psi \neq \psi_0 \) the physical model mismatch in the MSE cannot be compensated for by the NN, since it does not take \( u(t-n_k-1) \) as input).

**V. EXPERIMENTAL VALIDATION**

Effectiveness of the developed PGNN feedforward controllers is validated on the problem of closed-loop position control for the real-life coreless linear motor (CLM) in Fig. 2 that is also considered in [11]. Training data is generated in closed-loop for a duration of 120 s at a frequency of 1 kHz following (19), with \( C(q^{-1}) = \frac{9804q^{-1} - 1.946 \cdot 10^4 q^{-2} + 9651 q^{-3}}{1 - 2.348 q^{-1} + 2.492 q^{-2} - 0.753 q^{-3}} \), \( \Delta u(t) \sim \mathcal{N}(0, 50^2) \) and a third order reference \( r(t) \) that moves back and forth from \(-0.1\) to \( 0.1 \), with \( \max(|\dot{r}(t)|) = 0.05 \frac{m}{s} \), \( \max(|\ddot{r}(t)|) = 4 \frac{m}{s^2} \), and \( \max(|\dddot{r}(t)|) = 1000 \frac{m}{s^4} \).

Newton’s second law gives the physics-based model

\[
\ddot{y}(t) = -\frac{f_u}{m} \dot{y}(t) - \frac{f_r}{m} \text{sign}(\delta y(t)) + \frac{1}{m} u(t),
\]

with \( \delta \) a discrete–time differential operator, e.g., backward Euler \( \delta = \frac{1-q^{-1}}{q^{-1}} \), and \( m, f_u, f_r \) the mass, viscous friction coefficient, and Coulomb friction coefficient, respectively. Identification of the parameters in (30) according to (8) gives \( \theta_{\text{phys}}^* = \{m^*, f_u^*, f_r^*\} = \{21.01, 160.7, 7.386\} \).

We consider the following feedforward controllers:

1. **Direct inverse**, i.e., PGNN (14) trained according to (15). Essentially, the PGNN as proposed in [10].

2. **Indirect optimization–based inverse**, i.e., PGNN (11) trained according to (13), with the feedforward resulting from Algorithm 1 using \( k = 5 \) iterations.

3. **Indirect analytical inverse**, i.e., PGNN (24) trained according to (27) with feedforward (25).

All PGNNs are NARX models to facilitate a fair comparison with the direct inverse approach. Empirical tests resulted in the choice of one hidden layer with 16 tanh–neurons, see also [11]. Training is performed using Levenberg–Marquardt with \( \Lambda = 0.01I \). For each configuration, the PGNN was selected that achieved the smallest converged cost function out of 10 trainings with random weight initialization.

Fig. 3 shows the tracking error \( e(t) = r(t) - \dot{y}(t) \) for the PGNN feedforward controllers using a reference \( r(t) \) with \( \max(|\dot{r}(t)|) = 0.15 \frac{m}{s} \). The indirect methods proposed in this paper exhibit significantly smaller tracking errors, mostly during acceleration. Indeed, the indirect identification based PGNN feedforward controllers reduce the peak error, i.e., \( \max(|e(t)|) \), with a factor of more than three.
In this paper, we presented a framework for nonlinear feedforward control design in the presence of noisy data using physics–guided neural networks. First, by using fundamental knowledge from the system identification field, we formulate assumptions on the PGNN model, data, and training to obtain a consistent estimation of the forward system dynamics. Afterwards, two approaches are proposed for inversion of the identified PGNN to obtain the feedforward controller: an online numerical inversion, and an analytic inversion. The developed methodology was validated on a real–life industrial linear motor, where it significantly outperformed the direct inverse approach for the same PGNN.

VI. CONCLUSIONS

In this paper, we presented a framework for nonlinear feedforward control design in the presence of noisy data using physics–guided neural networks. First, by using fundamental knowledge from the system identification field, we formulate assumptions on the PGNN model, data, and training to obtain a consistent estimation of the forward system dynamics. Afterwards, two approaches are proposed for inversion of the identified PGNN to obtain the feedforward controller: an online numerical inversion, and an analytic inversion. The developed methodology was validated on a real–life industrial linear motor, where it significantly outperformed the direct inverse approach for the same PGNN.

| Table 1 |
| --- |
| Mean computation time per sample for the reference in Fig. 3 on a 2.59 GHz Intel Core–i7–9750H using MATLAB 2019a. |
| Dir. | Ind. optimization | Ind. analytical |
| Comp. time [s] | 4.18 · 10⁻⁷ | 1.28 · 10⁻⁴ | 3.84 · 10⁻⁷ |

REFERENCES

[1] S. Devasia, “Should model–based inverse inputs be used as feedforward under plant uncertainty?” IEEE Trans. on Automatic Control, vol. 47, pp. 1865–1871, 2002.

[2] M. L. G. Boerlage, M. Steinbuch, P. F. Lambrechts, and M. M. J. van de Wal, “Model–based feedforward for motion systems,” IEEE Int. Conf. on Control Applications, vol. 2, pp. 1158–1163, 2003.

[3] Z. Jamaludin, H. van Brussel, and J. Swevers, “Friction compensation of an XY feed table using friction–model–based feedforward and an inverse–model–based disturbance observer,” IEEE Trans. on Industrial Electronics, vol. 56, no. 10, pp. 3848–3853, 2009.

[4] K. Igarashi, R. Igarashi, T. Atsumi, and S. Nakadai, “Feedforward control for track–seeking control in hard disk drive with sampled–data polynomial based on first–order hold,” IEEE Int. Conf. on Mechatronics, 2021.

[5] T. T. Nguyen, M. Lazar, and H. Butler, “Cancellation of normal parasitic forces in coreless linear motors,” 19th Int. Conf. on StCC, pp. 192–199, 2015.

[6] O. Sørensen, “Additive feedforward control with neural networks,” IFAC Proc. Vol., vol. 32, no. 2, pp. 1378–1383, 1999.

[7] M.-G. Zhang, “Inverse model identification of nonlinear dynamic system using neural network,” ICMLC, vol. 5, pp. 2451–2455, 2007.

[8] C. Wanigasekara, A. Swain, S. K. Nguang, and P. Gangadhara, “Neural network based inverse system identification from small data sets,” Int. Joint Conf. on Neural Networks, pp. 1–6, 2019.

[9] K. Hornik, “Approximation capabilities of multilayer feedforward networks,” Neural Networks, vol. 4, pp. 251–257, 1991.

[10] M. Bolderman, M. Lazar, and H. Butler, “Physics–guided neural networks for inversion–based feedforward control applied to linear motors,” IEEE CCA, pp. 1115–1120, 2021.

[11] ——, “On feedforward control using physics–guided neural networks: Training cost regularization and optimized initialization,” ECC, 2022.

[12] D. Psaltis, A. Sideris, and A. A. Yamamura, “A multilayered neural network controller,” IEEE Control Systems Mag., vol. 8, no. 2, pp. 17–21, 1988.

[13] Y. Jung and M. Enqvist, “Estimating models of inverse systems,” 52nd IEEE CDC, pp. 7143–7148, 2013.

[14] K. J. Hunt, D. Sharbaro, R. Zbikowski, and P. J. Gawthrop, “Neural networks for control systems – a survey,” Automatica, vol. 28, no. 6, pp. 1083–1112, 1992.

[15] C. A. Jensen, R. D. Reed, R. J. Marks, M. A. El-Sharkawi, J. B. Jung, R. T. Miyamoto, G. M. Anderson, and C. J. Eggen, “Inversion of feedforward neural networks: Algorithms and applications,” Proc. of the IEEE, vol. 87, no. 9, pp. 1536–1549, 1999.

[16] L. Aarnoudse, W. Ohnishi, M. Poot, P. Tacx, N. Strijbosch, and L. Ljung, “On the consistency of prediction error identification methods,” Math. in Science and Engineering, vol. 126, pp. 121–164, 1976.

[17] M. Steinbuch and R. M. J. van de Molengraft, “Iterative learning control of industrial motion systems,” IFAC–PapersOnline, vol. 50, no. 1, pp. 12.101–12.106, 2017.

[18] T. T. Nguyen, M. Lazar, H. Butler, and P. M. J. van den Hof, “An instrumental variable method for closed–loop identification of coreless linear motors,” ACC, pp. 5219–5224, 2018.

[19] J. Kon, M. Heerjes, and T. Oomen, “Neural network training using closed–loop data: Hazards and an instrumental variable (IVNN) solution,” arXiv preprint arXiv:2202.05337, 2022.