In 61th IEEE Conference on Decision and Control, Cancun, Mexico, 2022

Unifying Model-Based and Neural Network Feedforward: Physics-Guided Neural Networks with Linear Autoregressive Dynamics

Johan Kon[1], Dennis Bruijnen[2], Jeroen van de Wijdeven[3], Marcel Heertjes[1,3], and Tom Oomen[1,4]

***Abstract*— Unknown nonlinear dynamics often limit the tracking performance of feedforward control. The aim of this paper is to develop a feedforward control framework that can compensate these unknown nonlinear dynamics using universal function approximators. The feedforward controller is parametrized as a parallel combination of a physics-based model and a neural network, where both share the same linear autoregressive (AR) dynamics. This parametrization allows for efficient output-error optimization through Sanathanan-Koerner (SK) iterations. Within each SK-iteration, the output of the neural network is penalized in the subspace of the physics-based model through orthogonal projection-based regularization, such that the neural network captures only the unmodelled dynamics, resulting in interpretable models.**

I. Introduction

Feedforward control can significantly increase the performance of dynamic systems [1], [2], e.g., positioning accuracy in motion systems. In feedforward control, the key requirements are high tracking performance and task flexibility [3], i.e., a small tracking error for a variety of references. Additionally, it is often desired that the feedforward controller is interpretable [4], and that its parameters can be efficiently learned given a training dataset.

Feedforward controllers based on physical models are highly flexible and interpretable by design [5]. For example, the dynamics can be parametrized as a rational transfer function [6] These parametrizations allow for efficient optimization [7] and can be interpreted through frequency-domain tools, e.g., Bode diagrams. Extensions include static friction [8] and position-varying compliance feedforward [9], as well as methods to compensate for nonminimum-phase zero dynamics [10]. However, these physics-based parametrizations often have limited performance in the presence of unknown, typically nonlinear dynamics [11], [12].

On the other hand, feedforward signals that compensate all reproducible dynamics, i.e., achieve tracking performance up to the noise level of the system, can be generated through learning control methods such as iterative learning control (ILC) [13]. Yet, these approaches lack task flexibility, necessitating the use of, e.g., basis functions [11], and do not result in interpretable feedforward signals.

To go beyond the trade-off between performance and task flexibility, universal function approximators such as neural networks have been used as flexible feedforward parametrizations [14], overcoming the performance decrease of physics-based parametrizations in the context of unmodelled dynamics. Examples include nonlinear autoregressive exogenous (NARX) and nonlinear finite impulse response (NFIR) parametrizations [15], [16], and long short-term memory neural networks [12]. As a downside, these parametrizations are not interpretable, and learning their parameters is computationally challenging. Additionally, these universal approximators lack the ability to extrapolate [4], deteriorating task flexibility outside the training regime.

Physics-guided neural networks (PGNNs) [17], [18] are a combined model-approximator parametrization and aim to reconcile the interpretability and task flexibility of model-based approaches with the performance of universal function approximators. Physics-guided parametrizations indeed significantly improve performance over model-based feedforward alternatives [19]. Interpretability is obtained through explicitly separating the neural network and model contribution by imposing orthogonality [20]. Even so, the performance of these PGNNs is limited as they do not contain AR dynamics and thus cannot compensate for zero dynamics of the system.

Although major steps have been taken to improve the flexibility of data-driven feedforward control while maintaining interpretability, at present these are limited by existing classes of PGNNs that can only handle overly simplified system dynamics. The aim of this paper, therefore, is to develop a class of PGNNs for feedforward control that can compensate zero dynamics. The main contribution is a PGNN feedforward control framework with AR dynamics, in which the model is interpretable and the neural network learns only unmodelled dynamics. This is achieved through the following subcontributions:

C1) A physics-guided feedforward parametrization with shared linear autoregressive dynamics (Section II).

C2) An efficient output-error optimization algorithm based on SK-iterations [7] (Section III and V).

C3) An orthogonal projection-based regularizer promoting orthogonality of the model and neural network, ensuring interpretability of the model (Section IV).

Notation and Definitions: All systems are discrete-time with sample time T_s. The sets $\mathbb{Z}_{>0}$, $\mathbb{R}_{\geq 0}$ represent the set of positive integers and non-negative real numbers. For the signal u with length N, $u(k) \in \mathbb{R}$ represents the signal at time index $k = \mathbb{Z}_{[1,N]}$, whereas $\underline{u} = \begin{bmatrix} u(1) & \dots & u(N) \end{bmatrix}^T \in \mathbb{R}^N$ is its finite-time vector representation. The set $\mathbb{R}[q^{-1}]$ is the set of polynomials in q^{-1} with real coefficients, with $q^{-1}u(k) = u(k-1)$. $\mathrm{Id}(\cdot)$ represents the identity operator.

This work is supported by Topconsortia voor Kennis en Innovatie (TKI), and ASML and Philips Engineering Solutions. [1]: Control Systems Technology Group, Departement of Mechanical Engineering, Eindhoven University of Technology, P.O. Box 513, 5600 MB Eindhoven, The Netherlands, e-mail: j.j.kon@tue.nl. [2]: Philips Engineering Solutions, High Tech Campus 34, 5656 AE Eindhoven, The Netherlands. [3]: ASML, De Run 6501, 5504 DR Veldhoven, The Netherlands. [4]: Delft University of Technology, P.O. Box 5, 2600 AA Delft, The Netherlands.

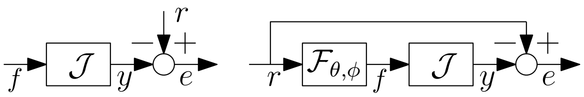

Fig. 1. Feedforward setup with input f, dynamic system $\mathcal{J}$, reference r, and error e (left). The input f is parametrized as the output of a reference dependent filter $\mathcal{F}_{\theta,\phi}$ (right).

II. Problem Formulation

In this section, first the problem of feedforward control for dynamic systems is introduced. Second, the physics-guided feedforward parametrization consisting of a physics-based model and neural network with shared linear AR dynamics is defined. Lastly, the learning problem is formulated.

A. Feedforward Setup and Physics-Guided Parametrization

The goal of feedforward control, see Fig. 1, is to generate input $f(k) \in \mathbb{R}$ to the discrete-time system $\mathcal{J}$ such that its output $y(k) \in \mathbb{R}$ equals the desired output $r(k) \in \mathbb{R}$, i.e.,

$$e(k) = r(k) - y(k) = r(k) - \mathcal{J}(f(k)) = 0 \quad \forall k \in \mathbb{Z}_{>0}, \tag{1}$$

with $e(k) \in \mathbb{R}$ the tracking error. The system $\mathcal{J}$ can represent a feedback-controlled or open-loop system.

To obtain both high performance and task flexibility, the input signal f is parametrized as the output of a feedforward controller acting on reference r. More specifically, the feedforward controller $\mathcal{F}_{\theta,\phi}$ is a parallel combination of a physics-based model $\mathcal{M}_\theta$ that is linear in its parameters (LIP) θ, and universal approximator $\mathcal{C}_\phi$ with parameters ϕ.

Definition 1 (Model class) *The model $\mathcal{M}_\theta : r(k) \rightarrow f_\mathcal{M}(k)$ satisfies the ordinary difference equation*

$$f_\mathcal{M}(k) + \sum_{i=1}^{N_b} b_i q^{-i} f_\mathcal{M}(k) = \sum_{i=1}^{N_a} a_i g_i \left(\psi_i(r(k))\right), \tag{2}$$

with $\psi_i \in \mathbb{R}[q^{-1}]$ and static nonlinearities $g_i : \mathbb{R} \rightarrow \mathbb{R}$, both user-defined functions, and parameters $\theta = \{a_i\}_{i=1}^{N_a} \cup \{b_i\}_{i=1}^{N_b}$, $a_i, b_i \in \mathbb{R}$.

Examples that can be encapsulated by this model class are, i.a., the class of rational transfer functions for $g_i = \mathrm{Id}$ and $\psi_i = q^{-i}$, and trigonometric nonlinearities resulting first-principles modelling, such as $a_i g_i(\psi_i(r(k)) = mgl\cos(\phi)$ for an inverse pendulum.

Definition 2 (Approximator class) *The approximator $\mathcal{C}_\phi : r(k) \rightarrow f_\mathcal{C}(k)$ satisfies the ordinary difference equation*

$$f_\mathcal{C}(k) + \sum_{i=1}^{N_b} b_i q^{-i} f_\mathcal{C}(k) = g_\phi(r(k)), \tag{3}$$

where $g_\phi(r(k))$ is the output of a neural network given by

$$\begin{array}{lll} h^l(r(k)) = \left[r(k), \ldots, r(k-q)\right]^T & \textit{if} & l = 0 \\ h^l(r(k)) = \sigma\left(W^{l-1}h^{l-1}(k) + c^l\right) & \textit{if} & l = 1, \ldots, L \\ g_\phi(r(k)) = W^l h^l(r(k)) & \textit{if} & l = L, \end{array} \tag{4}$$

with $W^l \in \mathbb{R}^{N_l \times N_{l-1}}$ the weights and $c^l \in \mathbb{R}^{N_l}$ the biases of layer l with n_l neurons, $\sigma(\cdot)$ an element-wise activation function, and parameter set $\phi = \{W^l, c^l\}_{l=0}^{L-1} \cup \{W^L\}$.

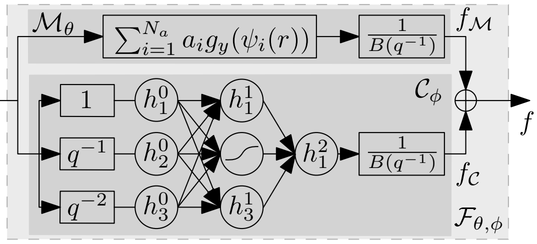

Fig. 2. Feedforward filter $\mathcal{F}_{\theta,\phi}$ as the parallel combination of model $\mathcal{M}_\theta$ and approximator $\mathcal{C}_\phi$ sharing AR dynamics $B(q^{-1}) = 1 + \sum_{i=1}^{N_b} b_i q^{-i}$, in this example with 2 hidden layers of 3 neurons and no skip connections.

The network $g_\phi(\cdot)$ acts on a past window of references $r(k)$, and is here represented by a fully connected multilayer perceptron without skip connections, see $\mathcal{C}_\phi$ in Fig. 2. It can be replaced by any network with a directed acyclic graph structure, e.g., residual neural networks [21], including user-defined input transformations and a bias in the final layer.

Since $\mathcal{M}_\theta$ and $\mathcal{C}_\phi$ share the same linear AR dynamics $f(k) + \sum_{i=1}^{N_b} b_i q^{-i} f(k)$, the parallel combination $\mathcal{F}_{\theta,\phi}$, see Fig. 2, also has these linear AR dynamics, as defined next.

Definition 3 *The feedforward controller $\mathcal{F}_{\theta,\phi} : r(k) \rightarrow f(k)$ is given by*

$$\mathcal{F}_{\theta,\phi}(r(k)) = \mathcal{M}_\theta(r(k)) + \mathcal{C}_\phi(r(k)), \tag{5}$$

such that it satisfies

$$\underbrace{(1 + \sum_{i=1}^{N_b} b_i q^{-i})}_{B(q)} f(k) = \underbrace{\sum_{i=1}^{N_a} a_i g_i(\psi_i(r(k)))}_{A(r(k))} + g_\phi(r(k)). \tag{6}$$

The parametrization $\mathcal{F}_{\theta,\phi}$ has nonlinear exogenous dynamics $A(r(k)) + g_\phi(r(k))$ and linear AR dynamics $B(q)f(k)$. Therefore, $\mathcal{F}_{\theta,\phi}$ is less complex than a NARX parametrization [12] with nonlinear AR dynamics, but it can capture a relevant class of physical systems with linear zero dynamics, as shown in Section VI, which cannot be captured by NFIR [20] or rational transfer function [11] parametrizations. In addition, the linear AR dynamics allow for linear stability analysis and inversion tools [22], and for efficient output-error (OE) minimization through SK-iterations [7].

To learn parameters θ, ϕ, a dataset $\mathcal{D} = \{r(k), \hat{f}(k)\}_{k=1}^N$ is assumed to be available with reference $r(k)$ and the corresponding input $\hat{f}$, such that $r(k) = \mathcal{J}(\hat{f}(k))$. This input $\hat{f}$ can be obtained by, e.g., ILC.

B. Problem Formulation

The aim of this paper is to learn parameters θ, ϕ of $\mathcal{F}_{\theta,\phi}$ in (6) based on dataset $\mathcal{D}$, such that $f(k) = \hat{f}(k)$, implying $e(k) = 0 \; \forall k \in \mathbb{Z}_{>0}$. This includes

1) an output error (OE) criterion that can be efficiently solved through SK-iterations because of the shared linear AR dynamics $B(q)f$,
2) regularizing this OE criterion with an orthogonal projection-based regularizer to promote unique coefficients θ, resulting in interpretable models, and
3) illustrating the approach on a two-mass-damper-spring system with Stribeck-like friction characteristics.

III. SK Iterations for Output Error Minimization

In this section, an output error criterion is introduced to be minimized by the learned parameters θ, ϕ of $\mathcal{F}_{\theta,\phi}$ in (6) (contribution C2). This criterion can be seen as a sequence of weighted least-squares problems, known a SK-iterations.

The OE criterion directly penalizes deviations of $f(k)$ from $\hat{f}(k)$ to ensure that $f(k) = \hat{f}(k)$, as defined next.

Definition 4 *Given feedforward parametrization* (6) *and dataset $\mathcal{D}$, the OE criterion $J_{OE} \in \mathbb{R}_{\geq 0}$ is given by*

$$J_{OE} = \sum_{k=1}^{N} \left(\hat{f}(k) - \frac{1}{B(q)} \left(A(r(k)) + g_\phi(r(k)) \right) \right)^2, \quad (7)$$

in which $(B(q))^{-1}(\cdot)$ represents a filtering operation.

Criterion (7) is linear in the parameters a of the exogenous dynamics, but nonlinear in the parameters b of the AR dynamics. As a result, (7) is nonconvex in b.

This nonconvexity in b can also be regarded as an a priori unknown weighting function of a least-squares problem. More specifically, (7) can be written as

$$J_{OE} = \sum_{k=1}^{N} \left(\tfrac{1}{B(q)} (B(q)\hat{f}(k) - A(r(k)) - g_\phi(r(k))) \right)^2. \quad (8)$$

Criterion J_{OE} in (8) is still nonlinear in parameters b due to the filtering term $(B(q))^{-1}$, but is linear in b in the term $B(q)\hat{f}(k)$. Thus, given the weighting function $(B(q))^{-1}$, the problem is linear in $\theta = \operatorname{col}(a, b)$. This motivates the following optimization algorithm for J_{OE}.

Algorithm 5 (SK-iterations for OE optimization)

Given parametrization (6) *with parameters a, b, ϕ, and dataset $\mathcal{D}$, set $j = 1$ and initialize a^0, b^0, ϕ^0 according to some strategy (e.g., a^0, b^0 as the best linear approximation, and ϕ^0 through Glorot initialization [23]). Then, iterate:*

(1) Given $B^{j-1}(q)$, determine a^j, b^j, ϕ^j as

$$a^j, b^j, \phi^j = \arg\min_{a,b,\phi} J_{OE}^j, \quad (9)$$

with $J_{OE}^j \in \mathbb{R}_{\geq 0}$ given by

$$J_{OE}^j = \sum_{k=1}^{N} \left(\frac{1}{B^{j-1}(q)} \left(B(q)\hat{f}(k) - A(r(k)) - g_\phi(r(k)) \right) \right)^2. \quad (10)$$

(2) Set $j = j+1$ and go back to (1) until convergence, e.g., until $a^j = a^{j-1}$, $b^j = b^{j-1}$, $\phi^j = \phi^{j-1}$.

The minimization (10) can be carried out through standard optimizers by differentiating through $(B^{j-1}(q))^{-1}$.

In (10) and Algorithm 5, $(B(q))^{-1}$ is interpreted as an a priori unknown weighting function that is iteratively adjusted over the iterations. Through iterating over j, it is aimed to recover (8) when $B^{j-1}(q) = B^j(q)$. Despite the lack of theoretical convergence guarantees and the nonconvexity of (7), practical use of this SK algorithm has shown good convergence properties [11], [24].

IV. Orthogonal Projection-Based Regularizer

Since all iterations of Algorithm 5 for optimizing J_{OE} in (7) are the same up to the weighting $(B^{j-1}(q))^{-1}$, the first iteration J_{OE}^1 is analyzed for the simplified setting in which only the last layer of g_ϕ in (4) is optimized. The optimum corresponding to this simplified problem is often non-unique due to the universal approximator characteristics of g_ϕ. In this paper, an orthogonal projection-based regularization is used to ensure that the optimum for the model coefficients θ is unique (contribution C3). This non-uniqueness directly applies to the full case (7).

A. Non-Uniqueness of First SK Iteration

If only the last layer of g_ϕ in (4) is optimized, g_ϕ is also LIP, such that the first SK-iteration can be written as a convex least-squares problem. The solution to this least-squares problem is often non-unique due to the universal approximation characteristics of g_ϕ. More specifically, consider criterion J_{OE}^1 in (10) with $B^0(q) = 1$ defined below.

Definition 6 *Given feedforward parametrization* (6) *and dataset $\mathcal{D}$, J_{OE}^1 with $B^0(q) = 1$ is given by*

$$J_{OE}^1 = \sum_{k=1}^{N} \left(B(q)\hat{f}(k) - A(r(k)) - g_\phi(r(k)) \right)^2. \quad (11)$$

Remark 7 *This criterion can be recognized as the equation error corresponding to feedforward parametrization* (6).

Consider now the case in which all hidden layers of g_ϕ in (4) are fixed, and only the output layer is optimized, i.e.,

$$g_\phi(r(k)) = h^L(r(k))^T \phi^T, \quad (12)$$

with $\phi = W^L \in \mathbb{R}^{1 \times N_\phi}$. For this setting, the approximator is also LIP, which allows to rewrite criterion (11) as follows.

Lemma 8 *Given an approximator structure* (12), *J_{OE}^1 in* (11) *can be represented as*

$$J_{OE}^1 = \| \underline{\hat{f}} - M\theta - H(\underline{r})^T \phi^T \|_2^2, \quad (13)$$

where $\theta = \begin{bmatrix} a^T & b^T \end{bmatrix}$ and $M = \begin{bmatrix} R & -\hat{F} \end{bmatrix}$ with

$$\begin{aligned} \underline{\hat{f}} &= \begin{bmatrix} \hat{f}(1) & \hat{f}(2) & \dots & \hat{f}(N) \end{bmatrix}^T \in \mathbb{R}^N \\ \hat{F} &= \begin{bmatrix} q^{-1}\underline{\hat{f}} & q^{-2}\underline{\hat{f}} & \dots & q^{-N_b}\underline{\hat{f}} \end{bmatrix} \in \mathbb{R}^{N \times N_b} \\ R &= \begin{bmatrix} g_1(\psi_1(\underline{r})) & \dots & g_{N_a}(\psi_{N_a}(\underline{r})) \end{bmatrix} \in \mathbb{R}^{N \times N_a} \\ H(\underline{r}) &= \begin{bmatrix} h^L(r(1)) & \dots & h^L(r(N)) \end{bmatrix} \in \mathbb{R}^{N_\phi \times N}, \end{aligned} \quad (14)$$

in which $\psi_i(\underline{r}) = \begin{bmatrix} (\psi_i(r))(1) & \dots & (\psi_i(r))(N) \end{bmatrix}^T \in \mathbb{R}^N$ and g_i applies elementwise.

Criterion (13) is a standard least-squares problem for which the solution is given by the pseudoinverse.

Lemma 9 *Given $\begin{bmatrix} M & H(\underline{r})^T \end{bmatrix} \in \mathbb{R}^{N \times N_\theta + N_\phi}$, the minimizer θ^*, ϕ^* of J_{OE}^1 in* (13) *is given by*

$$\theta^*, \phi^* = \arg\min_{\theta,\phi} J_{OE}^1 = \begin{bmatrix} M & H(\underline{r})^T \end{bmatrix}^+ \underline{\hat{f}} + \begin{bmatrix} v_\theta \\ v_\phi \end{bmatrix}, \quad (15)$$

for any $v = \begin{bmatrix} v_\theta^T & v_\phi^T \end{bmatrix}^T \in \mathbb{R}^{N_\theta + N_\phi}$ such that $v \in$ $\ker \begin{bmatrix} M & H(\underline{r})^T \end{bmatrix}$*, where $(\cdot)^+$ represents the pseudoinverse.*

Even though $\begin{bmatrix} M & H(\underline{r})^T \end{bmatrix}$ is tall, i.e., $N > N_\theta + N_\phi$, ker $\begin{bmatrix} M & H(\underline{r})^T \end{bmatrix}$ can be non-empty by two mechanisms. Before discussing these mechanisms, the following is assumed.

Assumption 10 *For tall $M \in \mathbb{R}^{N \times N_\theta}$, rank $M = N_\theta$.*

This assumption corresponds to a persistence of excitation condition for the model parametrization (2). For $g_i(\cdot) = \text{Id}(\cdot)$, i.e., for rational model parametrizations, this is equivalent to a non-zero spectrum of r at N_θ points [25]. Assumption 10 now allows for the following lemma.

Lemma 11 ker $\begin{bmatrix} M & H(\underline{r})^T \end{bmatrix}$ *is nonempty if and only if one of the following conditions is satisfied.*

P1) There exists v_ϕ for which $H(\underline{r})^T v_\phi = 0$, and $\begin{bmatrix} 0 & v_\phi^T \end{bmatrix}^T \in \ker \begin{bmatrix} M & H(\underline{r})^T \end{bmatrix}$.

P2) There exists a column $M_i \in \text{im } H(\underline{r})^T$. Consequently, there exists a v such that $\begin{bmatrix} M & H(\underline{r})^T \end{bmatrix} v = 0$.

The case $P1$ corresponds to overparametrization of g_ϕ, and only results in non-unique approximator coefficients ϕ, which do not need to be interpretable, and is thus of no concern. In the case of $P2$, g_ϕ can represent (parts of) the model due to its universal function approximator characteristicswhich can be present in practice [20]. In this case, the model coefficients θ are not unique.

B. Orthogonal Decomposition

An explicit expression describing the subspace in which θ is non-unique is obtained through splitting the criterion (13) into orthogonal subspaces, which are chosen as the model output space im M, and its orthogonal complement.

More specifically, given that M has full rank, it can be factorized through a singular value decomposition (SVD).

Lemma 12 *$M \in \mathbb{R}^{N \times N_\theta}$, $N > N_\theta$, can be factorized as*

$$M = \begin{bmatrix} U_1 & U_2 \end{bmatrix} \begin{bmatrix} \Sigma \\ 0 \end{bmatrix} V^T, \tag{16}$$

with $U_1 \in \mathbb{R}^{N \times N_\theta}$, $U_2 \in \mathbb{R}^{N \times N - N_\theta}$, $V \in \mathbb{R}^{N_\theta \times N_\theta}$ unitary matrices such that $U_1^T U_1 = I_{N_\theta}$, $U_1^T U_2 = 0$, $U_1 U_1^T + U_2 U_2^T = I_N$, and $\Sigma \in \mathbb{R}^{N_\theta \times N_\theta} = diag(\sigma_1, \ldots, \sigma_{N_\theta})$ with $\sigma_i > 0$ [26].

Consequently, the model response $M\theta$ can be written as

$$M\theta = U_1 \Sigma V^T \theta, \tag{17}$$

in which U_1 is a basis for the output space of M, and U_2 its orthogonal complement. This explicit basis allows to decouple criterion (13) into orthogonal subspaces.

Theorem 13 *Given factorization* (16)*, J_{OE}^1 in* (13) *can be written as*

$$J_{OE}^1 = \left\| \begin{bmatrix} U_1^T \hat{\underline{f}} \\ U_2^T \hat{\underline{f}} \end{bmatrix} - \begin{bmatrix} \Sigma V^T & U_1^T H(\underline{r})^T \\ 0 & U_2^T H(\underline{r})^T \end{bmatrix} \begin{bmatrix} \theta \\ \phi^T \end{bmatrix} \right\|_2^2. \tag{18}$$

This decoupling can be interpreted as projection into the model coefficient space and into its orthogonal complement. The entry $U_1^T H(\underline{r})^T \phi^T$ represents the contribution of the approximator expressed in the coordinates of model coefficients. Theorem 13 allows for the following result.

Corollary 14 *Given* (15)*, if a vector $v = \begin{bmatrix} v_\theta^T & v_\phi^T \end{bmatrix}^T$ exists such that $\begin{bmatrix} M & H(\underline{r})^T \end{bmatrix} v = 0$, then v satisfies*

$$\begin{bmatrix} \Sigma V^T & U_1^T H(\underline{r})^T \\ 0 & U_2^T H(\underline{r})^T \end{bmatrix} \begin{bmatrix} v_\theta \\ v_\phi \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \tag{19}$$

such that $H(\underline{r})^T v_\phi \in \ker U_2^T = (\text{im } U_2)^\perp = \text{im } U_1$, and

$$v_\theta = -(\Sigma V^T)^{-1} U_1^T H(\underline{r})^T v_\phi = -M^+ H(\underline{r})^T v_\phi. \tag{20}$$

The case where $H(\underline{r})^T v_\phi = 0$ for $v_\phi \neq 0$ corresponds to $P1$ of Lemma 11. In contrast, $H(\underline{r})^T v_\phi \neq 0$ and $H(\underline{r})^T v_\phi \in \text{im } U_1$ corresponds to $P2$, i.e., there exists a linear subspace in which both the model M and approximator $H(\underline{r})^T$ can capture the same effects. Corollary 14 expresses the relation between v_θ and v_ϕ for any v in this subspace, describing the directions in which θ is non-unique.

C. Orthogonal Projection-Based Regularizer

To obtain unique model coefficients θ, J_{OE}^1 in (13) is regularized with an orthogonal projection-based regularization that penalizes the approximator output $H(\underline{r})^T \phi^T$ in the subspace of the model $M\theta$. This orthogonal projection-based cost function for J_{OE}^1 where g_ϕ is LIP is defined next.

Definition 15 *Given dataset $\mathcal{D}$ and J_{OE}^1 in* (13)*, the criterion $J_{OE,P}^1 \in \mathbb{R}_{\geq 0}$ is defined as*

$$J_{OE,P}^1 = \|\hat{\underline{f}} - M\theta - H(\underline{r})^T \phi^T\|_2^2 + \lambda R(\phi), \tag{21}$$

in which $R(\phi) \in \mathbb{R}_{\geq 0}$ is given by

$$R(\phi) = \|(\Sigma V^T)^{-1} U_1^T H(\underline{r})^T \phi^T\|_2^2. \tag{22}$$

The regularizer $R(\phi)$ penalizes the *scaled* approximator output $H(\underline{r})^T \phi^T$ in im $M = \text{im } U_1$ through $U_1^T H(\underline{r})^T \phi^T$. Through the scaling $(\Sigma V^T)^{-1}$, $R(\phi)$ directly regularizes for $v_\theta = 0$, see (20). The structure of (22) allows for splitting (21) into orthogonal subspaces as formalized next.

Theorem 16 *Given factorization* (16)*, $J_{OE,P}^1$ in* (21) *can be written as*

$$J_{OE,P}^1 = \left\| \begin{bmatrix} U_1^T \hat{\underline{f}} \\ U_2^T \hat{\underline{f}} \\ 0 \end{bmatrix} - \begin{bmatrix} \Sigma V^T & U_1^T H(\underline{r})^T \\ 0 & U_2^T H(\underline{r})^T \\ 0 & \sqrt{\lambda} (\Sigma V^T)^{-1} U_1^T H(\underline{r})^T \end{bmatrix} \begin{bmatrix} \theta \\ \phi^T \end{bmatrix} \right\|_2^2. \tag{23}$$

Theorem 16 shows that the regularizer (22) adds additional rows to the decoupled optimization compared to (18) of Theorem 13. These extra rows ensure that unique model coefficients θ are recovered from $J_{OE,P}^1$, as illustrated next.

Corollary 17 *Given criterion J_{OE}^1 in* (18) *and $J_{OE,P}^1$ in* (23)*, nominal solution $\begin{bmatrix} M & H(\underline{r})^T \end{bmatrix}^+ \hat{\underline{f}} := x^*$, see Lemma 9, and any two vectors $v_1, v_2 \in \ker \begin{bmatrix} M & H(\underline{r})^T \end{bmatrix}$ such that $v_1 = \begin{bmatrix} 0 & v_\phi^T \end{bmatrix}^T$ and $v_2 = \begin{bmatrix} v_\theta^T & v_\phi^T \end{bmatrix}^T$ with $v_\theta \neq 0$, then,*

$$J_{OE}^1(x^* + v_1) = J_{OE}^1(x^* + v_2). \tag{24}$$

In contrast, for $J_{OE,P}^1$, it holds that

$$J_{OE,P}^1(x^* + v_1) < J_{OE,P}^1(x^* + v_2), \tag{25}$$

such that θ^* in $\arg\min_{\theta,\phi} J_{OE,P}^1$ is unique.

Corollary 17 conveys that the orthogonal projection-based regularizer (22) shrinks the non-unique directions v_θ to the zero vector: for any vector $\begin{bmatrix} v_\theta^T & v_\phi^T \end{bmatrix}^T \in \ker \begin{bmatrix} M & H(\underline{r})^T \end{bmatrix}$, the v_θ component is regularized to 0, such that unique model coefficients θ are recovered. Note that the contribution v_ϕ can still be non-unique, i.e., $P1$ of Lemma 11.

Remark 18 *Other regularization techniques,e.g., ℓ_2, could have also been employed to obtain unique θ in* (18)*. However, $R(\phi)$ in* (22) *only penalizes outputs of g_ϕ that can be captured by $M\theta$, whereas others also penalize outputs that can only be captured by g_ϕ, resulting in performance decrease.*

This section has shown that the optimum of J_{OE}^1 in (11) is non-unique already when only the last layer of g_ϕ in (4) is optimized. Naturally, this problem persists if all layers of g_ϕ are optimized, for which the linear subspace (20) becomes a complex nonlinear manifold in $\mathbb{R}^{N_\theta+N_\phi}$. Also in this full setting, $R(\phi)$ promotes unique θ for J_{OE}^1. This regularization is extended to subsequent SK-iterations in the next section.

V. Orthogonality at Each SK-Iteration

In this section, the orthogonal projection-based regularizer (22), is incorporated in the SK-iterations of Algorithm 5, see Section III, resulting in an efficient solver for OE minimization that promotes uniqueness of θ at each iteration.

This uniqueness is achieved through an iteration-varying orthogonal projection-based regularizer. This regularizer is obtained through constructing an orthogonal decomposition of the weighted model response alike to Lemma 12. Then, J_{OE}^j is regularized similarly to (21), such that it can be decoupled like (23) at each iteration. Here, due to space constraints, only the resulting algorithm is presented.

Algorithm 19 (SK-iterations for OE minimization with orthogonal projection-based regularization)

Given parametrization (6) *with parameters a, b, ϕ, and dataset $\mathcal{D}$, set $j = 1$ and initialize a^0, b^0, ϕ^0. Then, iterate:*

(1) *Given $B^{j-1}(q)$, calculate its convolution matrix W^{j-1} such that the finite-time response $y(k) = (B^{j-1}(q))^{-1}u(k)$ is given by $\underline{y} = W^{j-1}\underline{u}$ with*

$$W^{j-1} = \begin{bmatrix} h(0) & h(-1) & \dots & h(1-N) \\ h(1) & h(0) & \dots & h(2-N) \\ \vdots & & \ddots & \vdots \\ h(N-1) & h(N-2) & \dots & h(0) \end{bmatrix}, \tag{26}$$

with $h(k)$ the impulse response of $(B^{j-1}(q))^{-1}$.

(2) *Rewrite J_{OE}^j in* (10) *as a vector norm, i.e.,*

$$J_{OE}^j = \|W^{j-1}\left(\underline{\hat{f}} - M\theta - g_\phi(\underline{r})\right)\|_2^2. \tag{27}$$

(3) *Obtain the SVD of $W^{j-1}M$ as*

$$W^{j-1}M = \begin{bmatrix} U_1^{j-1} & U_2^{j-1} \end{bmatrix} \begin{bmatrix} \Sigma^{j-1} \\ 0 \end{bmatrix} V^{j-1^T}. \tag{28}$$

(4) *Construct iteration-varying orthogonal projection-based regularizer $R^{j-1}(\phi)$ as*

$$R^{j-1}(\phi) = \|(\Sigma^{j-1}V^{j-1^T})^{-1}U_1^{j-1^T}W^{j-1}g_\phi(\underline{r})\|_2^2. \tag{29}$$

(5) *Determine a^j, b^j, ϕ^j as*

$$a^j, b^j, \phi^j \arg\min_{a,b,\phi} = J_{OE}^j + \lambda R^{j-1}(\phi). \tag{30}$$

(6) *Set $j = j + 1$ and go back to (1) until convergence.*

In this regularized SK-algorithm, $R^{j-1}(\phi)$ directly promotes uniqueness of θ at each iteration through optimizing g_ϕ such that $W^{j-1}g_\phi(\underline{r}) \notin \operatorname{im} U_1^{j-1}$, and consequently $W^{j-1}M\theta$ captures all effects that can be encapsulated by the model. Thus, heuristically, θ is unique at convergence, resulting in unique model coefficients for (6).

VI. Simulation Example

In this section, feedforward parametrization (6) is validated on an example dynamic system that is contained in this parametrization. It is shown that it outperforms the feedforward class of rational transfer functions.

A. Example System

The dynamic system $\mathcal{J} : f(k) \to y(k)$ is given by a two-mass-spring-damper system, see Fig. 3, with a nonlinear damper d_{NL} connecting m_1 to the fixed world. The nonlinear damper represents Stribeck-like friction characteristics often found in stage systems for lithographic inspections tools, for which a simple model is given by

$$d_{NL}(\delta y(k)) = c_1\delta y(k) + \frac{c_2 - c_1}{\cosh(\alpha\delta y(k))}\delta y(k), \tag{31}$$

which is visualized in Fig. 4. The system parameters are given by $m_1 = 1$, $m_2 = 2$, $k_1 = 1$, $k_2 = 15000$, $d_2 = 50$, $c_1 = 1$, $c_2 = 20$, $\alpha = 20$, representing a stiff connection between m_1 and m_2, resulting in a high-frequency flexible mode.

For this system, a dataset of 9 references is generated combined with the optimal input $\hat{f}$ for each reference.

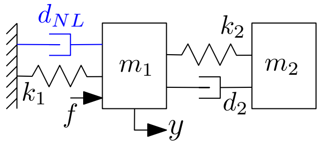

Fig. 3. Two-mass-damper-spring system with nonlinear Stribeck-like friction characteristics d_{NL}.

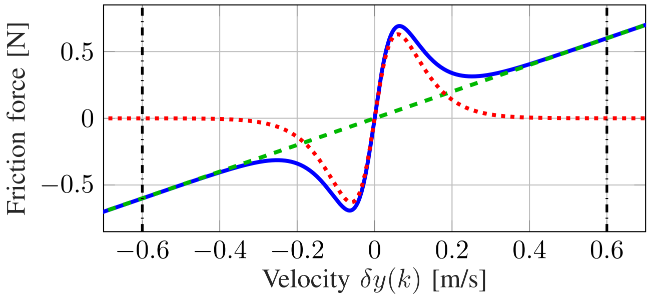

Fig. 4. Stribeck-like friction curve $d_{NL}(\delta y(k))$ (—) of example system in Fig. 3 with $c_1 = 1$, $c_2 = 20$, $\alpha = 20$, consisting of a linear (- -) and nonlinear (···) contribution.

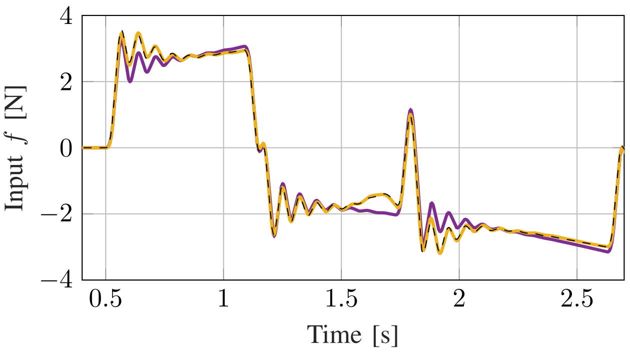

Fig. 5. The feedforward signal generated by the parallel parametrization $\mathcal{F}_{\theta,\phi}$ (—) is able to capture the optimal input $\hat{f}$ (- -) for which $e = 0$ up to approximation capabilities, resulting in $\|e\|_2^2 = 0.0104$. In contrast, the feedforward signal generated by a rational transfer function (—) is not able to correctly capture the nonlinear effects, resulting in $\|e\|_2^2 = 5.925$ m^2.

B. Performance Increase over Rational Basis Functions

Consider the following feedforward parametrizations.

1) A linear model $\mathcal{M}_\theta$ in (2) with $g_i = \text{Id}$, $\psi_i(r(k)) = q^{-i+1}r(k)$ and $N_a = 10$, $N_b = 9$, corresponding to a 10^{th} order rational transfer function. Note that this is an overparametrization of the linear part of Fig. 3.
2) A parallel parametrization $\mathcal{F}_{\theta,\phi}$ in (6) with $g_i = \text{Id}$, $\psi_i(r(k)) = q^{-i+1}r(k)$ and $N_a = 5$, $N_b = 2$, and g_ϕ with $L = 3$, $N_0 = 5$, $N_1 = 10$, $N_2 = 10$, $N_3 = 1$, i.e., the last 5 reference samples as input, 2 hidden layers and 1 output layer, with 10 neurons in each hidden layer. Note that this parametrization is able to capture the dynamics up to the approximation capabilities of g_ϕ.

Parametrization 1) is optimized according to criterion J_{OE} in (7) through SK-iterations, see Algorithm 5, whereas parametrization 2) is optimized with orthogonal projection-based cost function, see Algorithm 19. Fig. 5 shows the optimal input $\hat{f}$ and the generated input f of above parametrizations for a validation reference, resulting in errors $\|e\|_2^2 = 5.925$ m^2 for the rational transfer function, and $\|e\|_2^2 = 0.0104$ m^2 for $\mathcal{F}_{\theta,\phi}$. This illustrates that $\mathcal{F}_{\theta,\phi}$ is able to effectively capture the effect of the nonlinear damper $d_{NL}(\delta y(k))$, resulting in improved performance.

VII. Conclusion

This paper has developed a feedforward control framework that enables superior performance over model-based feedforward control, while maintaining interpretability and task flexibility. The feedforward controller is parametrized as a parallel combination of a physics-based model and neural network, with shared autoregressive dynamics, exactly encapsulating a class of nonlinear systems with linear zero dynamics. The physics-based model and neural network are optimized simultaneously according to an output-error criterion using SK-iterations. At each SK-iteration, complementarity of the physics-based model and neural network is promoted through an iteration-dependent orthogonal projection-based regularizer. This regularizer penalizes the output of the neural network in the subspace of the model, resulting in interpretable model coefficients. The superior performance of the framework over a rational feedforward parametrization is validated on a two-mass-damper-spring system with nonlinear friction characteristics.

References

[1] G. M. Clayton, S. Tien, K. K. Leang, Q. Zou, and S. Devasia, "A review of feedforward control approaches in nanopositioning for high-speed SPM," *J. Dyn. Syst. Meas. Control*, vol. 131 (6), 2009.

[2] L. R. Hunt, G. Meyer, and R. Su, "Noncausal inverses for linear systems," *IEEE Trans. Automat. Contr.*, vol. 41 (4), pp. 608–611, 1996.

[3] J. A. Butterworth, L. Y. Pao, and D. Y. Abramovitch, "A comparison of control architectures for atomic force microscopes," *Asian J. Control*, vol. 11 (2), pp. 175–181, 2009.

[4] J. Schoukens and L. Ljung, "Nonlinear System Identification: A User-Oriented Road Map," *IEEE Control Syst.*, vol. 39 (6), pp. 28–99, 2019.

[5] P. Lambrechts, M. Boerlage, and M. Steinbuch, "Trajectory planning and feedforward design for electromechanical motion systems," *Control Eng. Pract.*, vol. 13 (2), pp. 145–157, 2005.

[6] Q. Zou, "Preview-based stable-inversion for output tracking of linear systems," *Automatica*, vol. 45 (1), pp. 230–237, 2009.

[7] C. K. Sanathanan and J. Koerner, "Transfer function synthesis as a ratio of two complex polynomials," *IEEE Trans. Automat. Contr.*, vol. 8 (1), pp. 56–58, 1963.

[8] M. Boerlage, M. Steinbuch, P. Lambrechts, and M. Van De Wal, "Model-based feedforward for motion systems," in *Proc. Conf. Control Appl.*, vol. 2, 2003, pp. 1158–1163.

[9] N. Kontaras, M. Heertjes, H. Zwart, and M. Steinbuch, "A compliance feedforward scheme for a class of LTV motion systems," in *Proc. Am. Control Conf.*, 2017, pp. 4504–4509.

[10] S. Devasia, D. Chen, and B. Paden, "Nonlinear inversion-based output tracking," *IEEE Trans. Automat. Contr.*, vol. 41 (7), pp. 930–942, 1996.

[11] J. Bolder and T. Oomen, "Rational basis functions in iterative learning control - With experimental verification on a motion system," *IEEE Trans. Control Syst. Technol.*, vol. 23 (2), pp. 722–729, 2015.

[12] L. Ljung, C. Andersson, K. Tiels, and T. B. Schön, "Deep learning and system identification," *IFAC-PapersOnLine*, vol. 53 (2), 2020.

[13] D. A. Bristow, M. Tharayil, and A. G. Alleyne, "A survey of iterative learning control," *IEEE Control Syst. Mag.*, vol. 26 (3), 2006.

[14] K. J. Hunt, D. Sbarbaro, R. Żbikowski, and P. J. Gawthrop, "Neural networks for control systems—A survey," *Automatica*, vol. 28 (6), 1992.

[15] J. Sjöberg, Q. Zhang, L. Ljung, A. Benveniste, B. Delyon, P. Y. Glorennec, H. Hjalmarsson, and A. Juditsky, "Nonlinear black-box modeling in system identification: a unified overview," *Automatica*, vol. 31, no. 12, pp. 1691–1724, 1995.

[16] K. S. Narendra and K. Parthasarathy, "Identification and control of dynamical systems using neural networks," *IEEE Trans. Neural Networks*, vol. 1 (1), pp. 4–27, 1990.

[17] A. Karpatne, G. Atluri, J. H. 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," *IEEE Trans. Knowl. Data Eng.*, vol. 29 (10), pp. 2318–2331, 2017.

[18] A. Karpatne, W. Watkins, J. Read, and V. Kumar, "Physics-guided neural networks (PGNN): An application in lake temperature modeling," *arXiv*, 2017.

[19] M. Bolderman, M. Lazar, and H. Butler, "Physics-guided neural networks for inversion-based feedforward control applied to linear motors," *Conf. Control Technol. Appl.*, pp. 1115–1120, 2021.

[20] J. Kon, D. Bruijnen, J. van de Wijdeven, M. Heertjes, and T. Oomen, "Physics-guided neural networks for feedforward control: An orthogonal projection-based approach," in *Proc. Am. Control Conf.*, 2022.

[21] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pp. 770–778, 2016.

[22] Q. Zou, "Optimal preview-based stable-inversion for output tracking of nonminimum-phase linear systems," *Automatica*, vol. 45 (1), 2009.

[23] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Int. Conf. Artif. Intell. Stat.*, vol. 9. PMLR, 2010, pp. 249–256.

[24] A. H. Whitfield, "Asymptotic behaviour of transfer function synthesis methods," *Int. J. Control*, vol. 45 (3), pp. 1083–1092, 1987.

[25] L. Ljung, *System identification: theory for the user*, 2nd ed., T. Kailath, Ed. Prentice Hall PTR, 1999.

[26] D. C. Lay, *Linear algebra and its applications*. Pearson Education, 2003.