2023 62nd IEEE Conference on Decision and Control (CDC)
December 13-15, 2023. Marina Bay Sands, Singapore

Data–driven feedforward control design for nonlinear systems: A control–oriented system identification approach*

Max Bolderman[†,1], Mircea Lazar[1], and Hans Butler[1,2]

Abstract— **Feedforward controllers typically rely on accurately identified inverse models of the system dynamics to achieve high reference tracking performance. However, the impact of the (inverse) model identification error on the resulting tracking error is only analyzed a posteriori in experiments. Therefore, in this work, we develop an approach to feedforward control design that aims at minimizing the tracking error a priori. To achieve this, we present a model of the system in a lifted space of trajectories, based on which we derive an upperbound on the reference tracking performance. Minimization of this bound yields a feedforward control–oriented system identification cost function, and a finite–horizon optimization to compute the feedforward control signal. The nonlinear feedforward control design method is validated using physics–guided neural networks on a nonlinear, nonminimum phase mechatronic example, where it outperforms linear ILC.**

I. INTRODUCTION

Feedforward control is a dominant actor in achieving high reference tracking performance, and typically relies on linear, physics–based models [1], [2]. Linear models have good extrapolation properties, but limited accuracy. As such, it would be desirable to employ rich, nonlinear models for feedforward control that can learn the complete system dynamics from data [3].

A common approach to feedforward control design is inverse model–based feedforward, which generates the feedforward signal by passing the reference through a model of the inverse system dynamics, see, e.g., [1], [4]. When the model of the system is nonminimum phase, i.e., it has an unstable inverse, different methods are available to generate a stable feedforward controller, see, e.g., [4], [5]. These methods however, are not directly extendable to nonlinear feedforward controllers. Hence, a different approach is to formulate feedforward control as an optimization problem, where the goal is to minimize the norm of the difference between the reference and the model output. Within this category, it is possible to optimize the complete feedforward signal [6], [7], or to parameterize the feedforward signal as a function of time or the reference and optimize over the parameters [8], [9]. When the system performs a repetitive task, an iterative learning control (ILC) method can be used to minimize the tracking error based on the tracking error of previous repetitions by updating the feedforward input [10], the parameters of an inverse model [11], or both [12].

The aforementioned methods typically assume a known, physics–based model of the system. To account for unknown dynamics, data–driven techniques have been explored in combination with artificial intelligence, e.g., neural network (NN) models [13], physics–informed neural networks and physics–guided neural network (PGNN) models [14], other hybrid model structures [15], or Gaussian processes [16].

When performing the identification, i.e., fit the model to the data, the identification cost function should be relevant for the intended use of the model [17], [18]. Therefore, when identifying a model for feedforward control, the identification cost function should push model errors in a region where these errors least affect the tracking performance. This is not achieved when performing an identification of the inverse dynamics directly, which is generally adopted in nonlinear (including PGNN) feedforward control due to the non–invertibility of nonlinear models in general, see, e.g., [3], [14], [19]. In [20], a control–relevant identification cost function, which filters the inverse model error with a linear model of the process sensitivity, was proposed to mitigate this issue. Alternatively, in [21], the authors proposed an inversion method for (PG)NNs which opens up the path to perform the identification of the forward dynamics, but did not yet achieve a quantitative relation between the tracking error and the identification error. Such a quantitative relation is desired to have a relevant identification cost function for the feedforward control objective.

Motivated by the above observations, in this paper, we establish a quantitative link between the tracking error and the identification error. The main contributions of this paper are as follows:

1) A lifted formulation of the nonlinear feedforward control problem in the space of finite–length trajectories, which enables the derivation of an explicit upperbound on the norm of the tracking error;
2) A feedforward control–oriented identification cost function, which minimizes the upperbound on the tracking error, and hence, it minimizes the tracking error itself a priori, during the design stage;
3) A finite–horizon optimal feedforward control (FHOFC) formulation for a general class of nonlinear, possibly nonminimum phase MIMO systems which allows for specifying input, output, and state constraints.

The developed FHOFC problem can be solved iteratively, yielding an iterative learning scheme for nonlinear systems. We prove that this iterative learning FHOFC recovers linear ILC [10], which is another contribution of this work.

*This work is supported by the NWO research project PGN Mechatronics, project number 17973.

[†]Corresponding author: m.bolderman@tue.nl.

[1]Control Systems Group, Eindhoven University of Technology, Groene Loper 19, Eindhoven, 5612 AP, The Netherlands.

[2]ASML, De Run 6501, Veldhoven, 5504 DR, The Netherlands.

979-8-3503-0123-6/23/$31.00 ©2023 IEEE

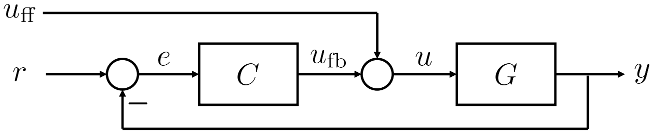

Fig. 1. Schematic overview of the control structure.

II. PRELIMINARIES

A. Notation

We denote $y(k) \in \mathbb{R}^{n_y}$ as the output at time $k \in \mathbb{N}_{>0}$, $r(k) \in \mathbb{R}^{n_y}$ is the reference, $e(k) := r(k) - y(k)$ the tracking error and $n_y \in \mathbb{Z}_{>0}$ the number of outputs. The input $u(k) \in \mathbb{R}^{n_u}$ is the sum of the feedback and the feedforward input, such that $u(k) := u_{\mathrm{fb}}(k) + u_{\mathrm{ff}}(k)$ with $n_u \in \mathbb{Z}_{>0}$ the number of inputs. The state of a system is denoted as $x(k) \in \mathbb{R}^{n_x}$ with $n_x \in \mathbb{Z}_{>0}$ the state dimension. A signal of length $N_k \in \mathbb{Z}_{>0}$ is denoted by its capital letter, e.g., $R := [r(1)^T, ..., r(N_k)^T]^T$ is the reference signal and $E := [e(1)^T, ..., e(N_k)^T]^T$ is the error signal. The superscript d is used to indicate that a signal is from the data set, e.g., $U_{\mathrm{ff}}^d = [u_{\mathrm{ff}}^d(1)^T, ..., u_{\mathrm{ff}}^d(N_d)^T]^T$ is the feedforward input measured during the data generating experiment of length $N_d \in \mathbb{Z}_{>0}$. Let a hat denote a prediction of a model, e.g., $\hat{Y} := [\hat{y}(1)^T, ..., \hat{y}(N_k)^T]^T$ is a prediction of the output Y and $\hat{Y}^d := [\hat{y}^d(1)^T, ..., \hat{y}^d(N_d)^T]^T$ a prediction of Y^d. A model is parametrized by the parameters $\theta \in \mathbb{R}^{n_\theta}$, $n_\theta \in \mathbb{Z}_{>0}$, and $\hat{\theta}$ denotes the identified parameters.

B. System dynamics and model–based feedforward

We consider the feedforward control design for a system G operating in closed–loop with feedback controller C as visualized in Fig. 1. The closed–loop system dynamics is (partly) unknown, which is the case in real–life systems, e.g., for a linear motor, one has to deal with parasitic effects, such as nonlinear friction and electromagnetic distortions. The closed–loop system dynamics is denoted as ϕ, such that

$$\phi : \begin{cases} x(k+1) &= f\big(x(k), u(k)\big), \\ y(k) &= g\big(x(k)\big), \\ u(k) &= C(q)\big(r(k) - y(k)\big) + u_{\mathrm{ff}}(k). \end{cases} \tag{1}$$

In (1), $f : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}^{n_x}$ describes the unknown system dynamics, with $g : \mathbb{R}^{n_x} \to \mathbb{R}^{n_y}$ the unknown output equation. The feedback controller is assumed to be linear, such that $u_{\mathrm{fb}}(k) = C(q)\big(r(k) - y(k)\big)$ with $C(q)$ the discrete–time transfer function of the feedback controller, and q the forward shift operator.

An input–output data set is generated by exciting the system via the reference $r(k)$ and the feedforward input $u_{\mathrm{ff}}(k)$, such that we obtain $Y^d = [y^d(1)^T, ..., y^d(N_d)^T]^T$, $R^d = [r^d(1)^T, ..., r^d(N_d)^T]^T$, and $U_{\mathrm{ff}}^d = [u_{\mathrm{ff}}^d(1)^T, ..., u_{\mathrm{ff}}^d(N_d)^T]^T$ that satisfies (1) for $k = 1, ..., N_d$.

The optimal feedforward input $u_{\mathrm{ff}}(k)$ yields $y(k) = r(k)$ for all k when supplied to the system (1). However, since f and g in (1) are unknown, it is common practice to parameterize a model $\hat{\phi}$ of the system ϕ in (1), such that

$$\hat{\phi} : \begin{cases} \hat{x}(k+1) &= \hat{f}\big(\theta, \hat{x}(k), \hat{u}(k)\big), \\ \hat{y}(k) &= \hat{g}\big(\theta, \hat{x}(k)\big), \\ \hat{u}(k) &= C(q)\big(r(k) - \hat{y}(k)\big) + u_{\mathrm{ff}}(k). \end{cases} \tag{2}$$

where $\hat{f}$ and $\hat{g}$ are a model of f and g in (1), respectively, and $\theta \in \mathbb{R}^{n_\theta}$ are the free parameters.

The state–space model in (2) reduces to the input–output representation used in, e.g., [14], [20] by choosing the state as past inputs and outputs, i.e.,

$$\begin{aligned} \hat{y}(k+1) = \hat{f}\big(\theta, [&\hat{y}(k)^T, ..., \hat{y}(k - n_a + 1)^T, \\ &\hat{u}(k - n_k)^T, ..., \hat{u}(k - n_k - n_b)^T]^T\big), \\ \hat{u}(k) = C(q)&\big(r(k) - \hat{y}(k)\big) + u_{\mathrm{ff}}(k), \end{aligned} \tag{3}$$

where $n_a, n_b, n_k \in \mathbb{Z}_{>0}$ denote the order of the dynamics. Suppose that there exists an inverse relation $\hat{f}^{-1}$ of $\hat{f}$ in (3), such that, with a slight abuse of notation, we have

$$\begin{aligned} \hat{u}(k) = \hat{f}^{-1}\big(\theta, [&\hat{y}(k + n_k + 1)^T, ..., \hat{y}(k + n_k - n_a + 1)^T, \\ &\hat{u}(k-1)^T, ..., \hat{u}(k - n_b + 1)^T]^T\big). \end{aligned} \tag{4}$$

Then, the inverse model–based feedforward controller is obtained by substitution of $\hat{y}(i) = r(i)$, $i = k + n_k - n_a + 1, ..., k + n_k + 1$, and $\hat{u}(i) = u_{\mathrm{ff}}(i)$, $i = k - n_b + 1, ..., k$.

III. PROBLEM FORMULATION

Since the dynamics f in (1) is unknown, the feedforward control design is based on a model $\hat{f}$ as in (2). This typically yields a two–step feedforward controller design procedure, consisting of an *identification* to fit the model (2) to the system (1) using the data $\{Y^d, R^d, U_{\mathrm{ff}}^d\}$, and an *inversion* to find the feedforward input U_{ff} for which the output $\hat{Y}$ of the model (2) follows the reference R.

The identification step aims to find the parameters $\theta = \hat{\theta}$ for the model $\hat{\phi}$ in (2) that best fit the data by minimizing a cost function, such that

$$\hat{\theta} = \arg\min_{\theta} V_{\mathrm{id}}(\theta, Y^d, R^d, U_{\mathrm{ff}}^d) + \|\Lambda(\theta - \theta^*)\|, \tag{5}$$

where $V_{\mathrm{id}} : \mathbb{R}^{n_\theta} \times \mathbb{R}^{n_y N_d} \times \mathbb{R}^{n_y N_d} \times \mathbb{R}^{n_u N_d} \to \mathbb{R}$ is the identification cost function, and $\Lambda \in \mathbb{R}^{n_\theta \times n_\theta}$ and $\theta^* \in \mathbb{R}^{n_\theta}$ are used for regularization. Suppose that $\hat{f}$ is a nonlinear input–output representation as in (3). Then, it is not generally possibly to find an inverse $\hat{f}^{-1}$ as in (4). A common approach to circumvent this issue is to parametrize a model $\hat{f}^{-1}$ directly, and identify its parameters using, e.g., a one step–ahead direct inverse identification, such that

$$\begin{aligned} V_{\mathrm{id}}(\theta, Y^d, R^d, U_{\mathrm{ff}}^d) &= \frac{1}{N} \sum_{i=1}^{N_d} \Big(u^d(k) - \hat{f}^{-1}\big(\theta, x^d(k)\big)\Big)^2, \\ x^d(k) = [y^d(k + n_k + 1), &..., y^d(k + n_k - n_a + 1), \\ &u^d(k-1), ..., u^d(k - n_b + 1)]^T. \end{aligned} \tag{6}$$

This approach is commonly adopted in literature, see, e.g., [3], [19], but fails to provide a quantitative relation

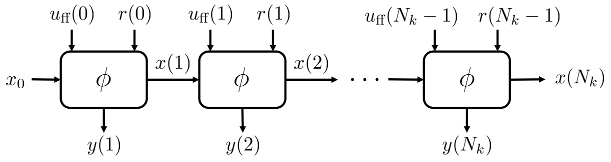

Fig. 2. Visual representation of the dynamics (1) in lifted form (7).

between the identification and the tracking error. Note that, the tracking error $e(k) = r(k) - y(k)$ does not even have the same unit as $u^d(k) - \hat{f}^{-1}\big(\theta, x^d(k)\big)$. In order to mitigate this issue, [20] proposes to filter the inverse model error by a linear model of the process sensitivity, i.e., minimize $G(q)S(q)\Big(u^d(k) - \hat{f}^{-1}\big(\theta, x^d(k)\big)\Big)$, with $G(q)$ and $S(q)$ the transfer function of the model of the system and the sensitivity, respectively. This approach establishes a qualitative relation between the tracking error and the identification cost function, but a quantitative relation is still missing.

Hence, in this work we will address two main issues: how to design the identification cost function V_{id} in (5), and how to compute the feedforward control input $u_{\text{ff}}(k)$ based on the identified nonlinear model, i.e., $\hat{\phi}$ in (2) with $\theta = \hat{\theta}$, such that the resulting tracking error is minimized for a general class of nonlinear, possibly nonminimum phase MIMO systems.

IV. FEEDFORWARD CONTROL–ORIENTED IDENTIFICATION

We rewrite the closed–loop system (1) and model (2) into a lifted form, based on which we propose the feedforward control–oriented identification cost function as well as the FHOFC formulation. Afterwards, we show that, indeed, minimizing the identification cost function and finding an optimal solution to the FHOFC minimizes the reference tracking error.

The lifted form of the closed–loop system is obtained by simulating ϕ in (1) as in Fig. 2, such that

$$Y = \Phi(x_0, R, U_{\text{ff}}). \tag{7}$$

In (1), $x_0 \in \mathbb{R}^{n_x}$ are the initial conditions, and $\Phi : \mathbb{R}^{n_x} \times \mathbb{R}^{n_y N_k} \times \mathbb{R}^{n_u \times N_k} \to \mathbb{R}^{n_y N_k}$ is a mapping obtained by recursive composition of the system ϕ. Similarly, the lifted form of the model $\hat{\phi}$ in (2) is defined as follows

$$\hat{Y} = \hat{\Phi}(\theta, x_0, R, U_{\text{ff}}), \tag{8}$$

where $\hat{\Phi} : \mathbb{R}^{n_\theta} \times \mathbb{R}^{n_x} \times \mathbb{R}^{n_y N_k} \times \mathbb{R}^{n_u N_k} \to \mathbb{R}^{n_y N_k}$ is a model–based mapping obtained by recursive composition of $\hat{\phi}$. The data set is generated by exciting the system with U_{ff}^d and R^d, such that, from (7) and (8), we can write

$$Y^d = \Phi(x_0^d, R^d, U_{\text{ff}}^d), \quad \hat{Y}^d = \hat{\Phi}(\theta, x_0^d, R^d, U_{\text{ff}}^d), \tag{9}$$

with $x_0^d \in \mathbb{R}^{n_\theta}$ the initial state of the experiment.

We aim to minimize the p–norm of the tracking error, i.e., $\|E\|_p = \|R - Y\|_p$ for some $p \in \mathbb{Z}_{\geq 1}$, while keeping the input, output, and states in the safe sets, i.e., $U \in \mathcal{R}_U$, $Y \in \mathcal{R}_Y$, and $X \in \mathcal{R}_x$. The feedforward control signal computation is done as follows:

1) **Identify** the optimal set of parameters $\hat{\theta}$ for the model $\hat{\phi}$ in (2) according to (5) with the feedforward control–oriented identification cost function

$$V_{\text{id}}(\theta, Y^d, R^d, U_{\text{ff}}^d) = \frac{1}{N_d^{1/p}} \|Y^d - \hat{\Phi}(\theta, x_0^d, R, U_{\text{ff}})\|_p. \tag{10}$$

2) **Compute** the feedforward input U_{ff} using the identified model, i.e., $\hat{\phi}$ with $\theta = \hat{\theta}$, according to the FHOFC

$$\begin{aligned} &U_{\text{ff}} = \arg\min_{U_{\text{ff}}} V_{\text{ff}}(\hat{\theta}, R, U_{\text{ff}}) + \|\Gamma U_{\text{ff}}\|, \\ &\text{subject to: } U_{\text{ff}} \in \mathcal{R}_{U_{\text{ff}}},\ \hat{U} \in \mathcal{R}_U,\ \hat{Y} \in \mathcal{R}_Y,\ \hat{X} \in \mathcal{R}_X. \end{aligned} \tag{11}$$

with the FHOFC cost function

$$V_{\text{ff}}(\hat{\theta}, R, U_{\text{ff}}) = \frac{1}{N_k^{1/p}} \|R - \hat{\Phi}(\hat{\theta}, x_0, R, U_{\text{ff}})\|_p. \tag{12}$$

Remark 4.1: The feedforward control–oriented identification cost function (10) penalizes the closed–loop simulation error of the model $\hat{\phi}$. This is different from the one–step–ahead inverse identification in (6), even when it is filtered with the process sensitivity. Moreover, [21] did not consider the feedback controller in the identification, and was therefore unable to link the tracking and identification error.

Remark 4.2: Solving the FHOFC optimization (11) becomes computationally expensive when N_k is large. However, the partial derivative of V_{ff} with respect to U_{ff} is known, such that, e.g., the constraint Gauss–Newton approach in [6] can be used. Several other options to reduce the computational complexity are: 1) *parametrize the feedforward signal* using basis functions [8], [9], 2) *Parametrize an inverse model* of the system and find its parameters via (11), or 3) *solve* (11) *in a receding horizon manner*.

Proposition 4.1: Consider the system ϕ in (1) with lifted form Φ in (7) and a corresponding parametrized model $\hat{\phi}$ in (2) with lifted form $\hat{\Phi}$ in (8). Suppose that $\hat{\theta}$ is identified according to (5) with V_{id} in (10), that U_{ff} is obtained from (11) with V_{ff} in (12), and define

$$\varepsilon := \frac{1}{N_k^{1/p}} \|Y - \hat{Y}\|_p - \frac{1}{N_d^{1/p}} \|Y^d - \hat{Y}^d\|_p. \tag{13}$$

Then, the tracking error resulting from U_{ff} satisfies

$$\frac{1}{N_k^{1/p}} \|R - Y\|_p \leq V_{\text{id}}(\hat{\theta}, Y^d, R^d, U_{\text{ff}}^d) + V_{\text{ff}}(\hat{\theta}, R, U_{\text{ff}}) + \varepsilon. \tag{14}$$

Proof: From the triangular inequality and (13), we have

$$\begin{aligned} \|R - Y\|_p &= \|R - \hat{Y} + \hat{Y} - Y\|_p \\ &\leq \|R - \hat{Y}\|_p + \|Y - \hat{Y}\|_p \\ &\leq \|R - \hat{Y}\|_p + \frac{N_k^{1/p}}{N_d^{1/p}} \|Y^d - \hat{Y}^d\|_p + N_k^{1/p}\varepsilon. \end{aligned} \tag{15}$$

Dividing both sides by $N_k^{1/p}$ and using V_{id} and V_{ff} as in (10) and (12) on the right hand side concludes (14). ■

The parameter ε in (13) is a measure stating the relevance of the *training data* $\{Y^d, R^d, U_{\text{ff}}^d\}$ with respect to the *operation data* $\{Y, R, U_{\text{ff}}\}$, which can be interpreted as the *validation data*. Accordingly, $\varepsilon > 0$ indicates that the training data does not sufficiently represent the operation data, while $\varepsilon < 0$ indicates that the training data covers all system dynamics that contribute to the tracking error. We aim for $\varepsilon = 0$, which is achieved when either:

1) *Training data* is the *operation data*, i.e., design $\{R^d, U_{\text{ff}}^d\} = \{R, U_{\text{ff}}\}$. An existing feedforward controller could be used for U_{ff}^d as an approximation of U_{ff};
2) *Consistent parameter identification*, such that $\Phi(x_0, R, U_{\text{ff}}) = \hat{\Phi}(\hat{\theta}, x_0, R, U_{\text{ff}})$ and $\Phi(x_0^d, R^d, U_{\text{ff}}^d) = \hat{\Phi}(\hat{\theta}, x_0^d, R^d, U_{\text{ff}}^d)$. This requires standard assumptions from system identification, i.e., the system is in the model set, the data is persistently exciting, and the optimization of (5) yields a global optimum, see, e.g., [21] for the formalized assumptions.

Suppose that the operation data has the same length as the data set, i.e., $N_k = N_d = N$ and that $x_0^d = x_0$. Then, with the triangular inequality, we upperbound ε in (13) by

$$\begin{aligned}\varepsilon &\le \frac{1}{N^{1/p}} \|Y - \hat{Y} - Y^d + \hat{Y}^d\|_p \\ &\le \frac{1}{N^{1/p}} \left\| \begin{bmatrix}\gamma_R, \gamma_{U_{\text{ff}}}\end{bmatrix} \begin{bmatrix} R - R^d \\ U_{\text{ff}} - U_{\text{ff}}^d \end{bmatrix} \right\|_p .\end{aligned} \tag{16}$$

where $\gamma_R = \max \left| \frac{\partial (Y - \hat{Y})}{\partial R} \right|$ and $\gamma_{U_{\text{ff}}} = \max \left| \frac{\partial (Y - \hat{Y})}{\partial U_{\text{ff}}} \right|$. Since Y and $\hat{Y}$ in (16) result from a closed–loop simulation, the feedback controller $C(q)$ also affects the upperbound in (16). Note that, for a linear system, $\frac{\partial Y}{\partial R}$ and $\frac{\partial Y}{\partial U_{\text{ff}}}$ describe the complementary sensitivity and process sensitivity, respectively.

Remark 4.3: Once the feedforward input U_{ff} is obtained from (11), it is possible to use it for generating new data which results in a smaller ε.

V. FINITE–HORIZON OPTIMAL FEEDFORWARD CONTROL

Next we show that the FHOFC (11) can recover some of the state–of–the–art methods for feedforward control, namely: inverse model–based feedforward and linear ILC. For simplicity, we neglect the constraints in (11).

Lemma 5.1: Consider the feedforward control design using an identified input–output model (3) for which the inverse relation (4) is unique. Suppose that the initial conditions are such that $y(i) = r(i)$ for $i \in \{n_k - n_a + 1, ..., n_k\}$. Then, the inverse model–based feedforward controller

$$\begin{aligned}u_{\text{ff}}(k) = f^{-1}\big(\hat{\theta}, [r^T(k + n_k + 1), ..., r^T(k + n_k - n_a + 1), \\ u_{\text{ff}}^T(k-1), ..., u_{\text{ff}}^T(k - n_b + 1)]^T\big)\end{aligned} \tag{17}$$

solves the unconstrained FHOFC (11) with $\Gamma = 0$ in a receding horizon manner with a preview of $n_k + 1$.

Proof: $u_{\text{ff}}(k)$ appears first in the predicted output $\hat{y}(k + n_k + 1)$, such that $u_{\text{ff}}(k + i)$, $i \in \mathbb{Z}_{>0}$ does not play a role in the cost function when $\Gamma = 0$. Hence, the FHOFC (11) with preview $n_k + 1$ becomes

$$u_{\text{ff}}(k) = \arg \min_{u_{\text{ff}}(k)} \|r(k + n_k + 1) - \hat{y}(k + n_k + 1)\|. \tag{18}$$

Since the minimum is attained for $r(k + n_k + 1) = \hat{y}(k + n_k + 1)$, the solution of (18) equals (17) when (18) is solved sequentially, i.e., for $k = 0, 1, ..., N_k$. Observing that, from (3), $\hat{u}(k) = C(q)\big(r(k) - y(k)\big) + u_{\text{ff}}(k) = u_{\text{ff}}(k)$ for $r(i) - \hat{y}(i) = 0$, $i \in \{0, ..., k\}$, completes the proof. ■

Remark 5.1: In [21], the optimization (18), which is a specific case of the FHOFC (11), was solved by proposing a specific PGNN structure for which f^{-1} of f in (3) is known analytically. Moreover, it discusses the use of a numerical solver when f^{-1} is not known.

When the identification in (5) and the FHOFC (11) are solved accurately, the tracking error may remain large due to ε in (14). To improve performance, it is possible to re–identify θ based on new data that is generated with U_{ff}. Alternatively, when the system has to perform a repetitive tasks, i.e., has to track a reference R multiple iterations, an iterative learning scheme can be implemented when the same reference is executed repetitively, as is done in ILC [10]. Inspired by ILC, we present the iterative learning FHOFC (IL–FHOFC) scheme:

$$\begin{aligned}\zeta &= \hat{\Phi}(\hat{\theta}, x_0, R, U_{\text{ff}}^{(i)}) + \alpha E^{(i)}, \\ U_{\text{ff}}^{(i+1)} &= Q_m \left[\arg \min_{U_{\text{ff}}} \|\zeta - \hat{\Phi}(\hat{\theta}, x_0, R, U_{\text{ff}})\| + \|\Gamma U_{\text{ff}}\| \right].\end{aligned} \tag{19}$$

In (19), the superscript (i) relates to iteration $i \in \mathbb{Z}_{>0}$ of a signal, $E^{(i)} = R - Y^{(i)}$ is the tracking error since R is constant over the iterations, $\zeta \in \mathbb{R}^{n_u N_k}$ is an auxiliary signal, and $Q_m \in \mathbb{R}^{n_u N_k \times n_u N_k}$ contains the Markov parameters of a robustness filter with transfer function matrix Q.

Lemma 5.2: Suppose that G is the transfer function matrix of a linear system, C is a feedback controller, and $\hat{G}$ is a model of G, with all transfer functions represented in the digital domain. Let $\hat{\Phi}$ be the closed–loop model–based mapping obtained from C and $\hat{G}$, and let $\alpha \in (0, 1]$ be a learning gain and Q a robustness filter. Then, the ILC law

$$\begin{aligned}\xi^{(i)}(k) &= u_{\text{ff}}^{(i)}(k) + \alpha \hat{G}^{-1}(1 + \hat{G}C)e^{(i)}(k), \\ u_{\text{ff}}^{(i+1)}(k) &= Q\xi^{(i)}(k),\end{aligned} \tag{20}$$

solves the IL–FHOFC problem (19) with $\Gamma = 0$.

Proof: The output predicted by $\hat{G}$ at iteration i is

$$\hat{y}^{(i)}(k) = (1 + \hat{G}C)^{-1}\hat{G}Cr(k) + (1 + \hat{G}C)^{-1}\hat{G}u_{\text{ff}}^{(i)}(k). \tag{21}$$

The output of the system with input $\xi^{(i+1)}(k)$ in (20) is

$$\hat{y}^{(i+1)}(k) = \hat{y}^{(i)}(k) + \alpha e^{(i)}(k). \tag{22}$$

Placing the time entries of (22) in a column, shows that

$$\left\| \zeta^{(i+1)} - \hat{\Phi}(\hat{\theta}, x_0, R, [\xi^{(i+1)}(1)^T, ..., \xi^{(i+1)}(N_k)^T]^T) \right\| = 0. \tag{23}$$

Hence, the first line of the linear ILC (20) solves the optimization in the IL–FHOFC (19) for $\Gamma = 0$. Since Q_m describes the robustness filter Q in ILC (20), $U_{\text{ff}}^{(i+1)}$ in (19) is the column of $u_{\text{ff}}^{(i+1)}(k)$ in (20). ■

Remark 5.2: Both Q_m and Γ in the IL–FHOFC (19) are parameters that affect convergence. Proving convergence for specific systems and models will be done in future work.

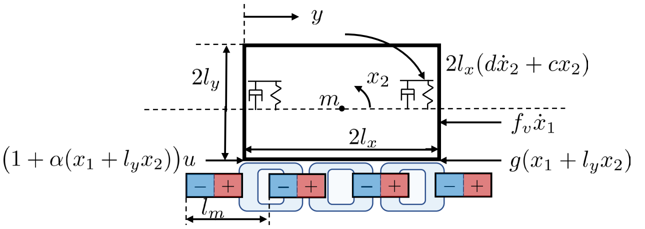

Fig. 3. Rotating–translating mass with actuation and sensing on opposite sides of the centre of mass.

TABLE I

PARAMETER VALUES OF THE SYSTEM DISPLAYED IN FIG. 3.

m	l_x, l_y	J	f_v	c	d	l_m	c_α	c_g
20	1	$\frac{40}{3}$	50	$\frac{25\cdot 10^3}{3}$	$\frac{575}{3}$	0.05	0.05	1
kg	m	kgm^2	$\frac{kg}{s}$	$\frac{kg}{s^2}$	$\frac{kg}{s}$	m	$-$	N

VI. VALIDATION ON A NONMINIMUM PHASE NONLINEAR MECHATRONIC EXAMPLE

System description: We consider the position control of a translating–rotating mass with force input u and position output y at opposite sides of the centre of mass, see Fig. 3. Let x_1 be the position of the centre of mass, and x_2 the rotation, such that u, x_1, x_2 and y are functions of time. Then, the continuous time dynamics are

$$\begin{aligned}\ddot{x}_1 &= \frac{1}{m}\Big(- f_v\dot{x}_1 + \big(1+\alpha(x_1+l_yx_2)\big)u \\ &\qquad - g(x_1+l_yx_2)\Big), \\ \ddot{x}_2 &= \frac{1}{J}\Big(- 2l_x(d\dot{x}_2 + cx_2) + \\ &\qquad l_y\big(1+\alpha(x_1+l_yx_2)\big)u - l_yg(x_1+l_yx_2)\Big), \\ y &= x_1 - l_yx_2.\end{aligned} \tag{24}$$

In (24), $l_x, l_y \in \mathbb{R}_{\geq 0}$ are the width and height of the mass $m \in \mathbb{R}_{>0}$, $J = \frac{1}{3}m(l_x^2 + l_y^2)$ is the moment of inertia, $f_v \in \mathbb{R}_{>0}$ the viscous friction coefficient, and $d, c \in \mathbb{R}_{>0}$ the damping and spring constant counteracting rotation from both ends of the mass. The nonlinearities $\alpha(x_1 + l_yx_2)$ and $g(x_1 + l_yx_2)$ represent the force ripple and cogging force, and are *unknown*. For simulation purposes, they are modelled as

$$\begin{aligned}\alpha(x_1+l_yx_2) &= c_\alpha \sin\big(\frac{2\pi}{l_m}(x_1+l_yx_2) + \frac{\pi}{4}\big), \\ g(x_1+l_yx_2) &= c_g \sin\big(\frac{2\pi}{l_m}(x_1+l_yx_2)\big),\end{aligned} \tag{25}$$

with $l_m \in \mathbb{R}_{>0}$ the magnet magnet pole pitch and $c_\alpha, c_g \in \mathbb{R}$ the riple and cogging magnitude. Parameter values are listed in Table I. The system (24) is controlled in closed–loop at a frequency $F_s = 100\ Hz$ by the ZOH discretization of

$$C(s) = 5\cdot 10^3 \frac{s+4\pi}{s+20\pi}. \tag{26}$$

A normally distributed noise $v(k) \sim \mathcal{N}\big(0, (10^{-6})^2\big)\ m$ is added as *measurement noise*. The system (24) exhibits the following challenges for feedforward control:

1) *Nonlinear dynamics* on the input (force ripple) and output (cogging force), depending on an internal state;
2) *Nonminimum phase dynamics* which requires non–causal actuation to return a stable feedforward signal.

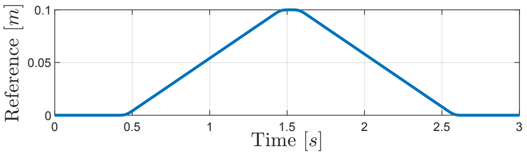

Fig. 4. Reference R used for performance evaluation.

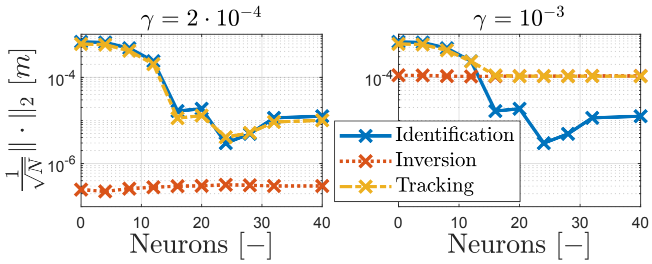

Fig. 5. Normalized 2–norm of the identification ($Y^d - \hat{Y}^d$, $N = N_d$), inversion ($R - \hat{Y}$, $N = N_k$) and tracking ($R - Y$, $N = N_k$) error for PGNNs with different number of neurons using $\gamma = 2\cdot 10^{-4}$ (left window) and $\gamma = 10^{-3}$ (right window).

Data generation: The training data is generated in closed–loop by sampling the output $y^d(k)$ at the frequency F_s for a duration of 45 s. The reference R^d is a concatenation of 15 times the third order reference R in Fig. 4, which has bounded velocity $|\frac{d}{dt}r^d| \leq 0.1\ \frac{m}{s}$, acceleration $|\frac{d^2}{dt^2}r^d| \leq 4\ \frac{m}{s^2}$ and jerk $|\frac{d^3}{dt^3}r^d| \leq 40\ \frac{m}{s^3}$. Additionally, U_{ff}^d is a zero–mean white noise with variance $\sigma^2 = 10^2\ N^2$ for $t = [10, 40)\ s$.

Model parametrization: We parameterize the system (24) with a state–space PGNN as

$$\begin{aligned}\dot{\hat{x}} &= A(\theta_{\text{phy}})\hat{x} + B(\theta_{\text{phy}})\big(\hat{u} + f_{\text{NN}}(\theta_{\text{NN}}, \hat{x}, \hat{u})\big), \\ \hat{y} &= C(\theta_{\text{phy}})\hat{x},\end{aligned} \tag{27}$$

where $\hat{u}$, $\hat{x}$ and $\hat{y}$ are functions of time, $f_{\text{NN}} : \mathbb{R}^{n_{\theta_{\text{nn}}}} \times \mathbb{R}^4 \times \mathbb{R} \to \mathbb{R}^4$ is a NN, and $\theta = [\theta_{\text{phy}}^T, \theta_{\text{NN}}^T]^T$ with θ_{phy} representing the physical parameters, and θ_{NN} the NN weights and biases. The NN has a single hidden layer with $n_1 \in \mathbb{Z}_{\geq 0}$ neurons, which we can vary. We discretize (27) using ZOH while assuming that f_{NN} remains constant in between two samples.

System identification: The PGNN parameters are identified according to (5) with feedforward control–oriented identification cost function (10) (using the *lsqnonlin* MATLAB function) with $\Lambda = 10^{-5}[\text{diag}(\theta_{\text{phy}}^T)^{-1}, 0]$, $\theta^* = [\theta_{\text{phy}}^{*\ T}, 0]^T$, and θ_{phy}^* the parameters obtained for the linear part of (27).

Feedforward: The feedforward signal U_{ff} has length $N_k = 300$ and is computed by solving the FHOFC (11) while penalizing the rate of change in U_{ff} by choosing $\Gamma = \gamma\Delta$, with $\gamma \in \mathbb{R}_{\geq 0}$ the importance of the regularization and $\Delta \in \mathbb{R}^{N_k \times N_k}$ has 1 on the diagonal, -1 on the subdiagonal and 0 elsewhere. Solving the FHOFC (11) converges in 7 s with *lsqnonlin* on a 2.59 GHz Intel Core–I7–9750H using MATLAB 2019A.

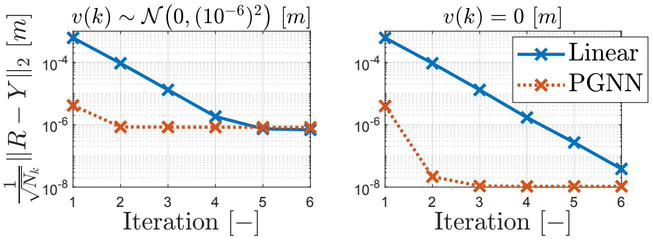

Fig. 6. Normalized 2–norm of the tracking error over the iterations using IL–FHOFC (19) with $\alpha = 1$, $\gamma = 10^{-5}$ and $Q_m = I$ using a linear and a PGNN (27) model with $n_1 = 24$, simulated with $v(k) \sim \mathcal{N}\big(0,(10^{-6})^2\big)$ (left window) and with $v(k) = 0$ (right window).

Results: Fig. 5 visualizes the 2–norm of the identification, inversion and tracking error in (14) for the reference R in Fig. 4 using different number of neurons n_1, with $\gamma = 2 \cdot 10^{-4}$ (left window) and $\gamma = 1 \cdot 10^{-3}$ (right window). Increasing the number of neurons n_1 in the model (27) improves the accuracy of the identification. For $\gamma = 2 \cdot 10^{-4}$ the inversion error is small, such that the tracking error is limited by the accuracy of the identification. In contrast, for $\gamma = 10^{-3}$ the inversion error increases, which limits the achievable performance for $n_i > 12$. Comparing the upperbound (14) with the results in Fig. 5 indicates that ε is small. Correspondingly, hyperparameters can be tuned based on (14), e.g., Fig. 5 shows that at least $n_l = 16$ neurons are required to achieve $\frac{1}{\sqrt{N_k}}\|E\|_2 < 10^{-4}$ m.

Fig. 6 shows the 2–norm of the tracking error for a linear and a PGNN model over multiple iterations of the reference R in Fig. 4 when using the IL–FHOFC (19). Since both approaches reach the noise–floor, results are added where $v(k) = 0$ to emphasize the benefit of the nonlinear PGNN model structure. The IL–FHOFC (19) with nonlinear PGNN model yields a significant reduction in the number of iterations required to reach a target performance.

VII. CONCLUSIONS

In this paper, we presented a generalized approach to nonlinear data–driven feedforward control design from the perspective of minimizing tracking errors. We showed that the norm of the reference tracking error is upperbounded by the sum of the inversion and the identification error, respectively. This resulted in a two–step approach to feedforward control design, consisting of a feedforward control–oriented system identification followed by a finite–horizon optimization to compute the feedforward input signal. Generality of the FHOFC formulation was demonstrated by recovering inverse model–based feedforward and linear ILC for specific settings.

REFERENCES

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

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

[3] M. Bolderman, H. Butler, S. Koekebakker, E. van Horssen, R. Kamidin, T. Spaan-Burke, N. Strijbosch, and M. Lazar, “Physics–guided neural networks for feedforward control with input–to–state stability guarantees,” *arXiv:2301.08568*, 2023.

[4] J. A. Butterworth, L. Y. Pao, and D. Y. Abramovitch, “Analysis and comparison of three discrete–time feedforward model–inverse control techniques for nonminimum–phase systems,” *Mechatronics*, vol. 22, no. 5, pp. 577–587, 2012.

[5] J. van Zundert and T. Oomen, “On inversion-based approaches for feedforward and ILC,” *Mechatronics*, vol. 50, pp. 282–291, 2018.

[6] M. Volckaert, A. van Mulders, J. Schoukens, M. Diehl, and J. Swevers, “Model based nonlinear iterative learning control: A constrained Gauss–Newton approach,” *Mediterranean Conference on Control and Automation*, pp. 718–723, 2009.

[7] D. S. Carrasco and G. C. Goodwin, “Feedforward model predictive control,” *Annual Review in Control*, vol. 35, pp. 199–206, 2011.

[8] K. S. Ramani, M. Duan, C. E. Okwudire, and A. G. Ulsoy, “Tracking control of linear time–invariant nonminimum phase systems using filtered basis functions,” *Journal of Dynamic Systems, Measurement, and Control*, vol. 139, p. 011001, 2017.

[9] R. Kasemsinsup, R. Romagnali, M. F. Heertjes, S. Weiland, and H. Butler, “Reference–tracking feedforward control deisgn for linear dynamical systems through signal decomposition,” *American Control Conference*, pp. 2387–2382, 2017.

[10] D. A. Bristow, M. Tharayil, and A. G. Alleyne, “A survey of iterative learning control,” *IEEE Control Systems Magazine*, vol. 26, no. 3, pp. 96–114, 2006.

[11] L. Blanken, F. Boeren, D. Bruijnen, and T. Oomen, “Batch–to–batch rational feedforward control: From iterative learning to identification approaches, with application to a wafer stage,” *IEEE/ASME Transactions on Mechatronics*, vol. 22, no. 2, pp. 826–837, 2017.

[12] M. B. Saltik, B. Jayawardhana, and A. Cherukuri, “Iterative learning and model predictive control for repetitive nonlinear systems via koopman operator approximation,” *IEEE Conference on Decision and Control*, pp. 3059–3065, 2022.

[13] O. Sørensen, “Additive feedforward control with neural networks,” *IFAC Proceedings Volumes*, vol. 32, no. 2, pp. 1378–1383, 1999.

[14] M. Bolderman, M. Lazar, and H. Butler, “Physics–guided neural networks for inversion–based feedforward control applied to linear motors,” *IEEE Conference on Control Technology and Applications*, pp. 1115–1120, 2021.

[15] C.-H. Chou, M. Dua, and C. E. Okwudire, “A physics–guided data–driven feedforward tracking controller for systems with unmodeled dynamics–applied to 3d printing,” *IEEE Access*, vol. 11, pp. 14 563–14 574, 2023.

[16] J. van Hulst, M. Poot, D. Kostić, K. W. Yan, J. Portegies, and T. Oomen, “Feedforward control in the presence of input nonlinearities: A learning–based approach,” *IFAC–PapersOnline*, vol. 55, no. 37, pp. 235–240, 2022.

[17] P. M. J. van den Hof and R. J. P. Schrama, “Identification and control – closed–loop issues,” *Automatica*, vol. 31, no. 12, pp. 1751–1770, 1995.

[18] J. Schoukens and L. Ljung, “Nonlinear system identification: a user–oriented road map,” *IEEE Control Systems*, vol. 39, no. 6, pp. 28–99, 2019.

[19] 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,” *American Control Conference*, pp. 4377–4382, 2022.

[20] L. Aarnoudse, W. Ohnishi, M. Poot, P. Tacx, N. Strijbosch, and T. Oomen, “Control–relevant neural networks for intelligent motion feedforward,” *IEEE International Conference on Mechatronics*, 2021.

[21] M. Bolderman, M. Lazar, and H. Butler, “Physics–guided neural networks for feedforward control: From consistent identification to feedforward controller controller design,” *IEEE Conference on Decision and Control*, pp. 1497–1502, 2022.