



HAL
open science

Identification for feedforward control of Wiener systems

Alain Uwadukunze, Xavier Bombois, Marco Forgone, Marion Gilson, Marie
Albisser

► **To cite this version:**

Alain Uwadukunze, Xavier Bombois, Marco Forgone, Marion Gilson, Marie Albisser. Identification for feedforward control of Wiener systems. 2024. hal-04495430

HAL Id: hal-04495430

<https://hal.science/hal-04495430>

Preprint submitted on 8 Mar 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Identification for Feedforward Control of Wiener Systems

UWADUKUNZE Alain^{*,****} BOMBOIS Xavier^{**}
FORGIONE Marco^{***} GILSON Marion^{*}
ALBISSER Marie^{****}

^{*} *Department of Control, Identification and Diagnostics, Université de Lorraine, CNRS, CRAN, F-54000 Nancy, France*

^{**} *Laboratoire Ampère, Ecole Centrale de Lyon, Université de Lyon, 36 avenue Guy de Collongue, Ecully, France and CNRS, France*

^{***} *Dalle Molle Institute for Artificial Intelligence, IDSIA USI-SUPSI, Via la Santa 1, CH-6962 Lugano-Viganello, Switzerland*

^{****} *French-German Research Institute, ISL, 5 rue du Général Cassagnou, 68300 Saint-Louis, France*

Abstract:

This paper presents an identification approach for feedforward controller design dedicated to nonlinear systems when their structure is known. We propose a model-based approach, leveraging Block-Oriented models, which allows a fast identification of the system when its structure is known. The controller is then designed based on the identified model using a criterion close to traditional identification criteria. Additionally, a costless iterative model identification and controller design approach is also proposed to improve the performance of the controller if necessary. The effectiveness of the approach is demonstrated through the design of a feedforward controller for a Wiener system.

Keywords: Identification for control, Identification, Nonlinear systems, Feedforward control, Block-oriented models

1. INTRODUCTION

From mechanical, electronic to aerospace systems, numerous real life applications exhibit nonlinear characteristics, posing significant challenges in the design of controllers. In this paper, we are particularly interested in a specific type of control: Feedforward control. The primary goal of this paper is to present an iterative identification procedure for the design of a feedforward controller for nonlinear systems which can be represented by Wiener structures (i.e., a cascade of a linear dynamic block and static nonlinearity).

Feedforward control is generally used to achieve the tracking of a given reference signal $r(t)$. This is generally achieved by determining the input signal that has to be applied to the true system using optimal control (Vinter and Vinter, 2010) or iterative learning control (ILC) (Bristow et al., 2006). In optimal control, the optimal input is determined based on a model of the true system while in ILC this input is determined using multiple experiments on the system (model-free approach). In both cases, the optimal input is determined for a given realization of the reference signal. This means that, for a reference having the same frequency content, but a different time realization, the whole procedure has to be repeated. This drawback in fact only pertains to the case of nonlinear systems. Indeed, in the linear case, we can, e.g., use H_2 control to design a feedforward controller \mathcal{F} (whose... expression depends on the frequency content of the reference signal). Once the controller \mathcal{F} has been designed for a given type of

reference, it can therefore be used to generate the optimal control input for different realizations of the reference (and for references having similar frequency content). It would therefore be interesting to be able to design such feedforward controllers also for nonlinear systems. This is what is proposed e.g. in (Aarnoudse et al., 2021), (Van Hulst et al., 2022) and (Poot et al., 2023). In these papers, iterative learning control is used to determine the optimal input for a representative realization of the reference signal and a nonlinear feedforward controller which allows to reproduce the optimal input when fed with the reference signal is then determined.

The ILC based approaches in (Aarnoudse et al., 2021), (Van Hulst et al., 2022) and (Poot et al., 2023) are particularly interesting when we do not have a precise idea of the structure of the true system. In this paper, we consider the situation where this structure is known. In particular, we will suppose that the true system is a Wiener system, i.e., the true system is made up of a linear transfer function followed by a static nonlinearity. In this case, one can use a single experiment on the true system to derive input-output data that can then be used to identify a model of the true system. Based on this identified model and a representative realization of the reference signal, we propose an approach to determine the feedforward controller using an approach close to an identification criterion. The structure of the feedforward controller is chosen as a Hammerstein system, i.e., a static nonlinearity followed by a linear transfer function. This

choice is motivated by the fact that feedforward control can be seen as a way to somehow invert the true system ((see also (Van Hulst et al., 2022) and (Poot et al., 2023)).

While iterative learning control generally requires multiple experiments on the true system, the identification approach proposed in the previous paragraph has the advantage that it requires in theory only one well-designed experiment on the true system in order to identify the model of the true system. It is clear though that, if this experiment is not well designed, the identified model will not be an accurate representation of the true system and the resulting feedforward controller will not be able to track the reference in a satisfactory manner. In this situation, the input-output data obtained by applying the feedforward controller on the true system can be used to improve the accuracy of the model and to update the feedforward controller. This procedure can be repeated until a suitable controller is synthesized. Note that this approach can be considered as costless since the additional data are obtained in a feedforward control configuration. Note that this iterative procedure can also be considered as an extension to the case of feedforward controller design for nonlinear (Wiener) systems of the classical iterative identification-control design procedure presented in (Van Den Hof and Schrama, 1995), (Gevers, 2002) and (Forgione et al., 2015) for the linear case.

As mentioned above, in this paper, the structure of the true system is assumed to be a Wiener system and the feedforward controller will have a Hammerstein structure. The Wiener and Hammerstein structures are classical examples of block-oriented models (Giri and Bai, 2010). This type of models allows to represent nonlinear systems by dividing them into interconnected static nonlinear and linear dynamic blocks. Due to their flexibility and easy-understanding, block-oriented models have found applications in the identification of numerous systems (Firouz et al., 2020), (Muhammad et al., 2021), (Naitali et al., 2008). They have also been used as models for to-be-controlled systems in, e.g., (Rollins et al., 2016), (Dolanc and Strmenik, 2008), and (Kalafatis et al., 2005). As indicated above, we will identify a Wiener model of the Wiener true system and we will design the Hammerstein feedforward controller using an approach close to an identification criterion. For this purpose, we can leverage the numerous identification approaches that are available for such model structures (see, e.g., (Schoukens et al., 2015) and (Schoukens et al., 2012)). In this paper, we will in particular take advantage of the recently introduced "dynoNet" nonlinear identification tool proposed in (Forgione and Piga, 2021).

To sum up, our goal in this paper will be to:

- Show how a feedforward controller can be designed from an identified block-oriented Wiener model of the to be controlled system.
- Present a costless iterative model identification and controller design approach in a case where an initially designed controller does not perform well on the true system.

The rest of this paper is structured as follows: first, the problem will be presented in Section 2, then the structures and identification approaches of the model and

the controller are presented in Section 3. The iterative procedure is presented in Section 4. Lastly, the procedures are illustrated through numerical examples in Section 5.

2. FRAMEWORK

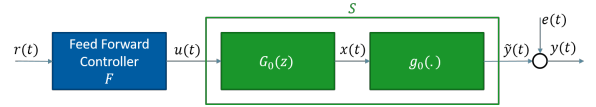


Fig. 1. Control scheme

In this study, we are interested in the design of a feedforward controller for the open loop control scheme presented in Figure 1. In this figure, $r(t)$ is the reference that the true system has to track. Like in the linear case, it is important to define the class of reference signals that we wish to track. In this paper, we will suppose that the frequency content of $r(t)$ can be described by a known transfer function $H(z)$. In other words, we will suppose that $r(t)$ is given by:

$$r(t) = H(z)\tilde{e}(t), \quad (1)$$

where $\tilde{e}(t)$ is a white Gaussian noise whose variance defines the amplitude range of $r(t)$. The signal $u(t)$ on Figure 1 is the input provided to the system and $y(t)$ is the measured output of the system.

The true system \mathcal{S} is considered to be a Wiener system. These systems are composed of a linear time dynamic block followed by a static nonlinearity. In this paper, the linear block, noted $G_0(z)$, is considered to be stable. The static nonlinearity is noted $g_0(\cdot)$.

In Wiener systems, the input $u(t)$ is filtered through the linear dynamic block $G_0(z)$, producing an intermediate signal $x(t) = G_0(z)u(t)$. This signal is then passed through the static nonlinearity $g_0(\cdot)$, yielding the output signal $\tilde{y}(t) = g_0(x(t))$. In the remainder of the paper, we will use the following expression for $\tilde{y}(t)$:

$$\tilde{y}(t) = g_0(G_0(z)u(t)). \quad (2)$$

In this paper, the output of the Wiener system $\tilde{y}(t)$ is corrupted with a white gaussian noise $e: \mathcal{N}(0, \sigma^2)$. For a given input $u(t)$, the measured output $y(t)$ can thus be expressed as follows:

$$y(t) = g_0(G_0(z)u(t)) + e(t). \quad (3)$$

The goal of this paper will be to design a feedforward controller \mathcal{F} for the system \mathcal{S} . For a given reference $r(t)$, the controller should provide a command $u(t)$ to pass to the system, allowing it to track the reference. The controller \mathcal{F} should be designed in order to minimize the following criterion:

$$E(r(t) - \tilde{y}(t))^2 + \lambda E u^2(t), \quad (4)$$

where E represents the expectation operator and $\lambda > 0$ is a user chosen constant which defines the penalty on the control efforts.

As indicated in the introduction, the controller \mathcal{F} must somehow achieve an inversion of the true system system. Since this true system has a Wiener structure, it therefore makes sense to choose a Hammerstein structure for the controller \mathcal{F} , i.e., a static nonlinearity followed by a linear dynamic block. The static nonlinearity of the feedforward controller will be denoted by $f(\cdot)$ and the linear dynamic block by $F(z)$. In Hammerstein systems the input $r(t)$ passes through the static nonlinearity $f(\cdot)$ providing an intermediate signal $z(t) = f(r(t))$. This signal is then filtered through the linear dynamic block $F(z)$ producing the output of the Hammerstein system $u(t) = F(z)z(t)$. In the remainder of the paper, we will therefore use the following expression for the control input $u(t)$ generated by the reference $r(t)$:

$$u(t) = F(z)f(r(t)). \quad (5)$$

In this paper, it is considered that an initial dataset $D_1 = \{u_1(t), y_1(t)\}_{t=1}^N$, obtained from an open-loop experiment on \mathcal{S} is available. Since we know the structure of the true system, we can use this dataset to identify a model \mathcal{M} of \mathcal{S} and then design the controller based on that model.

Since we know that the true system \mathcal{S} belongs to the class of Wiener systems, then the model \mathcal{M} will also have the same structure. The model \mathcal{M} will be identified using the data D_1 in a model structure parametrized by a parameter vector θ . This vector θ contains the parameters of both the linear block $G_\theta(z)$ and the static nonlinearity $g_\theta(\cdot)$.

In the next section, we will show how the model of the true system can be identified and how the feedforward controller can be determined based on this identified model and a representative realization of the reference signal (1). The latter will be done using a criterion which is very close to an identification criterion. Consequently, in the sequel, we will say (with some abuse of terminology) that both the Wiener model \mathcal{M} of the true system and the Hammerstein feedforward controller \mathcal{F} will be identified.

3. STRUCTURES AND IDENTIFICATION CRITERIONS

In this section, the identification procedures for \mathcal{M} and \mathcal{F} , as well as their structures are discussed. In this paper we choose a parametric identification approach for the Wiener and Hammerstein structures.

3.1 Structures and identification of \mathcal{M}

As said before, the model of the true system \mathcal{M} will have the exact same structure as the true system \mathcal{S} , i.e., a Wiener structure. It is parametrized as follows:

$$G_\theta(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2} + \dots + b_{n_b} z^{-n_b}}{1 + a_1 z^{-1} + a_2 z^{-2} + \dots + a_{n_a} z^{-n_a}} \quad (6a)$$

$$g_\theta(\cdot) = \tilde{g}_\theta(\cdot, \theta_g), \quad (6b)$$

where $\tilde{g}_\theta(\cdot, \theta_g)$, is a class of non linear static function described by its parameter vector $\theta_g \in \mathbb{R}^{n_g}$. We note by $A \in \mathbb{R}^{n_a}$ the vector containing the n_a denominators

of the linear block of \mathcal{M} and $B \in \mathbb{R}^{n_b}$ the vector containing the n_b numerators. $A = [a_1, a_2, \dots, a_{n_a}]^T$ and $B = [b_0, b_1, b_2, \dots, b_{n_b}]^T$. We note $\theta \in \mathbb{R}^{n_a + n_b + n_g}$ the vector containing all the parameters of the model \mathcal{M} , i.e., $\theta = [A^T, B^T, \theta_g^T]^T$.

The goal of the identification of \mathcal{M} is to find the parameters θ of \mathcal{M} which allow to represent accurately the behaviour of the true system. Given the initial dataset $D_1 = \{u_1(t), y_1(t)\}_{t=1}^N$, this can be achieved by minimizing the following cost function W :

$$W(\theta) = \frac{1}{N} \sum_{t=1}^N (g_\theta(G_\theta(z)u_1(t)) - y_1(t))^2, \quad (7)$$

where N is the number of samples collected. The parameter $\hat{\theta}$ which allow to best represent the model given a dataset $D_1 = \{u_1(t), y_1(t)\}_{t=1}^N$, is thus provided by:

$$\hat{\theta} = \arg \min_{\theta} W(\theta). \quad (8)$$

In the sequel, the blocks of the identified model \mathcal{M} (i.e., the one parametrized with $\hat{\theta}$) will be denoted by $\hat{G}_\theta(z)$ and $\hat{g}_\theta(\cdot)$. For a given input $u(t)$, the output \hat{y} of \mathcal{M} is therefore given by:

$$\hat{y}(t) = \hat{g}_\theta(\hat{G}_\theta(z)u(t)). \quad (9)$$

3.2 Structure and design of \mathcal{F}

The feedforward controller \mathcal{F} will have a Hammerstein structure described by a parameter vector ϕ . Its static non linearity, denoted by $f_\phi(\cdot)$, and linear dynamic layer, denoted by $F_\phi(z)$, are described as follows:

$$F_\phi(z) = \frac{d_0 + d_1 z^{-1} + d_2 z^{-2} + \dots + d_{n_d} z^{-n_d}}{1 + c_1 z^{-1} + c_2 z^{-2} + \dots + c_{n_c} z^{-n_c}} \quad (10a)$$

$$f_\phi(\cdot) = \tilde{f}_\phi(\cdot, \phi_f), \quad (10b)$$

where \tilde{f}_ϕ , is a class of non linear static function described by its parameter vector $\phi_f \in \mathbb{R}^{n_f}$. We note by $C \in \mathbb{R}^{n_c}$ the vector containing the n_c denominators of the linear block and $D \in \mathbb{R}^{n_d}$ the vector containing the n_d numerators. $C = [c_1, c_2, \dots, c_{n_c}]^T$ and $D = [d_0, d_1, d_2, \dots, d_{n_d}]^T$. We note $\phi \in \mathbb{R}^{n_c + n_d + n_f}$ the vector containing all the parameters of the controller: $\phi = [C^T, D^T, \phi_f^T]^T$.

We will suppose that we have a (representative¹ realization of duration N_r of the reference signal (1)) and we will approximate the control criterion (4) using this realization and the identified model $\hat{G}_\theta(z), \hat{g}_\theta(\cdot)$ of the true system:

$$V(\phi, \hat{\theta}) = \frac{1}{N_r} \sum_{t=1}^{N_r} \left(\hat{g}_\theta(\hat{G}_\theta(z)u_\phi(t)) - r(t) \right)^2 + \lambda u_\phi^2(t), \quad (11)$$

where $u_\phi(t) = F_\phi(z)f_\phi(r(t))$.

¹ i.e., a sufficiently long realization of (1)

Given a model $\hat{G}_\theta(z), \hat{g}_\theta(\cdot)$ the optimal parameter vector $\hat{\phi}$ is thus:

$$\hat{\phi} = \arg \min_{\phi} V(\phi, \hat{\theta}). \quad (12)$$

Note the similarity between the criterion (11)-(12) and the identification criterion (7)-(8). Similar optimization algorithms can therefore be used to determine both the model of the true system and the feedforward controller. This is also why we say in this paper that the feedforward controller is "identified".

The optimal controller which minimize the criteria (11) with parameter vector $\hat{\phi}$ will be denoted by $\hat{F}_{\hat{\phi}}(z)\hat{f}_{\hat{\phi}}(\cdot)$.

Remark In this paper the static nonlinearities of the model $g_\theta(\cdot)$ and the feedforward controller $f_\phi(\cdot)$, will be parametrized using a neural network structure (Forgione and Piga, 2021). Neural networks are used because they are known to excel in modeling complex static functions due to their flexibility. In the numerical examples presented in section 5, the neural networks used will have 1 input, 1 hidden layer and 1 output. However, more complex structures may be used if the latter do not allow to identify the static nonlinearity. In networks with 1 hidden layer, the output $f(x)$ is expressed as an affine combination of neurons $\nu_k(x)_{k=1}^{n_\nu}$ (the number of neurons is denoted n_ν):

$$f(x) = \sum_{k=1}^{n_\nu} w_k \nu_k(x) + b, \quad (13)$$

with b a scalar offset and w_k scalar weightings. The quantity $\nu_k(x)_{k=1}^{n_\nu}$ is a nonlinear mapping of the entry x :

$$\nu_k(x) = \Phi(\tilde{b}_k + \tilde{w}_k x), \quad (14)$$

where $\Phi(x)$ is generally the so-called activation function (typically *RELU* or *tanh*) and where \tilde{w}_k and \tilde{b}_k are scalar coefficients.

To implement the block-oriented models and to identify their parameters (i.e. to solve (7) and (11)), we use the *dynaNet* structure and tools presented in (Forgione and Piga, 2021).

4. ITERATIVE LEARNING PROCEDURE

Using the procedure presented in the previous section, a feedforward controller is available and this feedforward controller (say \mathcal{F}_1) can be used to design the control input of the true system using the reference signal $r(t)$. If the performance of the controller is not satisfactory (i.e., the output $y(t)$ is not sufficiently close to $r(t)$), the data generated during the application of the feedforward controller \mathcal{F}_1 can be used to improve the model. These data will be denoted by $D_2 = \{u_2(t), y_2(t)\}_{t=1}^N$ in the sequel². A new and better model \mathcal{M}_2 can be identified

using this new data set D_2 as well as the previous data D_1 . The identification criterion for the model \mathcal{M}_2 will be:

$$W_2(\theta) = \frac{1}{N} \sum_{t=1}^N (g_\theta(G_\theta(z)u_1(t)) - y_1(t))^2 + \frac{1}{N} \sum_{t=1}^N (g_\theta(G_\theta(z)u_2(t)) - y_2(t))^2. \quad (15)$$

The new parameter vector, $\hat{\theta}_2$, which allows to best represent the true system \mathcal{S} , given the dataset D_1 and D_2 is thus given by:

$$\hat{\theta}_2 = \arg \min_{\theta} W_2(\theta). \quad (16)$$

When a new model \mathcal{M}_2 has been identified, then a new controller \mathcal{F}_2 can also be designed based on \mathcal{M}_2 and using the identification criterion (11). The sequence consisting in first identifying a new model and then redesigning the controller based on that model can be repeated iteratively until a controller which performs well on the true system is found. As mentioned in the introduction, the learning of a better model can be considered as costless since the new data sets are generated when operating the plant in a control configuration. The procedure is described in Algorithm 1.

Algorithm 1 Iterative procedure

Initialize: Available dataset $D_1 = \{u_1(t), y_1(t)\}_{t=1}^N$ and a representative realization of length N_r of the to-be-tracked reference $r(t)$.

Repeat i_{max} **times the following steps**, $i = 1, \dots, i_{max}$

1. Identify a model \mathcal{M}_i using the data contained in the set $\mathcal{D} = \{D_1 = (u_1(t), y_1(t)), D_2 = (u_2(t), y_2(t)), \dots, D_i = (u_i(t), y_i(t))\}$ and the criterion:

$$W_i(\theta) = \frac{1}{N} \sum_{t=1}^N (g_\theta(G_\theta(z)u_1(t)) - y_1(t))^2 + \frac{1}{N} \sum_{t=1}^N (g_\theta(G_\theta(z)u_2(t)) - y_2(t))^2 + \dots + \frac{1}{N} \sum_{t=1}^N (g_\theta(G_\theta(z)u_i(t)) - y_i(t))^2.$$

2. Design the feedforward controller \mathcal{F}_i using the identified model \mathcal{M}_i and the design criterion:

$$V(\phi, \hat{\theta}_i) = \frac{1}{N_r} \sum_{t=1}^{N_r} \left(\hat{g}_{\hat{\theta}_i}(\hat{G}_{\hat{\theta}_i}(z)u_\phi(t)) - r(t) \right)^2 + \lambda u_\phi^2(t).$$

3. Perform an experiment on the true system \mathcal{S} using the designed controller \mathcal{F}_i , collect the data $D_{i+1} = \{u_{i+1}(t), y_{i+1}(t)\}_{t=1}^N$ and add them to the set \mathcal{D} .

End Repeat

5. EXAMPLES

In this section the procedures described above will be illustrated. To evaluate the goodness of the identification

² For the simplicity of notations, we will suppose that D_1 and D_2 contain the same amount of data

of the model and of the tracking of the reference by the controller, we use the "fit" and Root Mean Squared Error (RMSE) metrics:

$$fit_M = 100 \left(1 - \frac{\sqrt{\sum_{t=1}^{\tilde{N}} (\hat{y}(t) - y(t))^2}}{\sqrt{\sum_{t=1}^{\tilde{N}} (\hat{y}(t) - \bar{y}(t))^2}} \right) \% \quad (17a)$$

$$fit_F = 100 \left(1 - \frac{\sqrt{\sum_{t=1}^{\tilde{N}_r} (y(t) - r(t))^2}}{\sqrt{\sum_{t=1}^{\tilde{N}_r} (y(t) - \bar{r}(t))^2}} \right) \%, \quad (17b)$$

where \bar{a} represents the mean value of the vector containing the N observations of a . fit_M and fit_F are the fit on the identification of the model (computed using validation data, i.e., data that have not been used for the identification) and the fit on the tracking of the reference by the true system, respectively. The RMSE is defined as follows:

$$RMSE_M = \sqrt{\frac{1}{\tilde{N}} \sum_{t=1}^{\tilde{N}} (\hat{y}(t) - y(t))^2} \quad (18a)$$

$$RMSE_F = \sqrt{\frac{1}{\tilde{N}_r} \sum_{t=1}^{\tilde{N}_r} (y(t) - r(t))^2}, \quad (18b)$$

where $RMSE_M$ and $RMSE_F$ are the RMSE on the identification of the model (computed using the validation data) and the RMSE on the tracking of the reference by the true system respectively. Note that, since this is a simulation example, we could have used $\tilde{y}(t)$ instead of $y(t)$ in (17) and (18). The choice to use $y(t)$ instead of $\tilde{y}(t)$ is nevertheless made since $\tilde{y}(t)$ is unavailable in practice.

Two cases will be considered:

- In the first one, we will consider a case where there is no need for an iterative procedure to design the controller.
- In the second one, we will perform the iterative procedure (see Section 4).

For the 2 cases considered, we will use the same true system \mathcal{S} . As mentioned before, the true system is a Wiener system, i.e., it is composed by a linear dynamic block $G_0(z)$ followed by a static nonlinearity $f_0(x)$.

The linear dynamic bloc $G_0(z)$ of the true system \mathcal{S} is:

$$G_0(z) = \frac{1z^{-1}}{1 - 0.1412z^{-1} - 0.1z^{-2}}. \quad (19)$$

The static nonlinearity of the true system \mathcal{S} is:

$$f_0(x) = 5 \tan^{-1}(2x). \quad (20)$$

5.1 Controller synthesis without an iterative procedure

We will first show that, if the open-loop experiment leading to D_1 is well designed (i.e., u_1 is sufficiently broadband and sufficiently powerful), the identified model \mathcal{M}_1 is

sufficient to design a satisfactory controller \mathcal{F}_1 (no iterative procedure is required).

We assume here, that the following experiment has been conducted on the true system \mathcal{S} in order to collect a dataset D_1 :

- The system is excited with a multisine input $u_1(t)$ of maximum frequency of 2000 Hz.
- 20000 samples from the output of the system $y_1(t)$ are collected.
- The output is corrupted with a white gaussian noise of standard deviation $\sigma_e = 5.10e^{-4}$.

This dataset is divided into 2 sets of 10000 samples each. The first set is used to identify the model, the second one for validation.

The structure of the model is the one described in (6), where $n_a = 3, n_b = 4$ and \tilde{g}_θ is a neural network with 1 hidden layer (Equations (13) - (14)) with 16 neurons and a *tanh* activation function. The parameter vector has dimension 42. The optimal parameter vector $\hat{\theta}$ can be identified using the criteria (7)-(8).

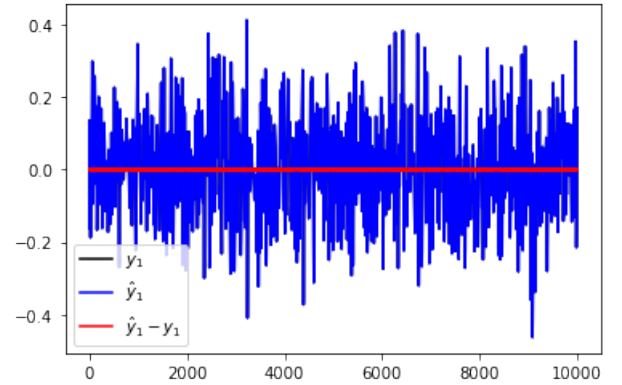


Fig. 2. Model output \hat{y}_1 (blue), true system output y_1 (black) and difference between \hat{y}_1 and y_1 (red) for a given input $u_1(t)$

Figure 2 presents the difference between the predicted output from the identified model \mathcal{M} compared to the output of the true system \mathcal{S} , on the validation set. As can be seen from this figure, the blue signal (\hat{y}_1) and the black signal (y_1) are overlapping and the red signal ($\hat{y}_1 - y_1$) is very small. The $RMSE_M = 6.0 \cdot 10^{-4}$ and the $fit_M = 99\%$. This shows that we are able to identify a parameter vector $\hat{\theta}$ that allows to represent well the model with the given data.

Now that a model has been identified from the data, a controller \mathcal{F}_1 can be designed from the model. In this case we want to design our controller to allow the model to follow a reference signal $r_a(t)$ which corresponds to a filtered white gaussian noise (1) where $H(z)$ is a lowpass filter with a cutoff frequency at 2000Hz:

$$H(z) = \frac{0.0014 + 0.004z^{-1} + 0.004z^{-2} + 0.001z^{-3}}{1 - 2.51z^{-1} + 2.13z^{-2} - 0.61z^{-3}}, \quad (21)$$

and where the variance of $\tilde{e}(t)$ has been chosen in such a way that the maximal amplitude of $r_a(t)$ is close to the

one of $y_1(t)$ in D_1 . The structure of the controller is the one described in (10), where $n_d = 4, n_c = 4$ and \hat{f}_ϕ is a neural network with 1 hidden layer, 16 neurons and a \tanh activation function. In total there are 44 parameters stored in the parameter vector ϕ to identify. These parameters are identified using the criteria (11)-(12) with $\lambda = 0$.

Once the controller has been designed from the model it can be tested on the true system as well.

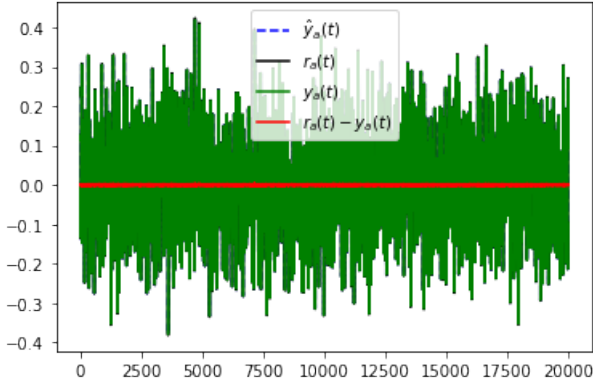


Fig. 3. Model output \hat{y}_a (blue), true system output y_a (green), reference signal r_a (black) and difference between r_a and y_a (red) when the input is the control signal provided by the designed controller \mathcal{F}_1

Figure 3 shows the outputs $y_a(t)$ and $\hat{y}_a(t)$ obtained when we test the identified controller on the true system and the model, respectively.

$$\hat{y}_a(t) = \hat{g}_\theta(\hat{G}_\theta(z)u_\phi(t)) \quad (22a)$$

$$y_a(t) = g_0(G_0(z)u_\phi(t)) + e(t), \quad (22b)$$

where $u_\phi(t) = \hat{F}_\phi(z)\hat{f}_\phi(r_a(t))$. As can be seen in Figure 3, $r_a(t)$, $y_a(t)$ and $\hat{y}_a(t)$ are all overlapping. This means that the designed controller not only performs well on the model but also on the true system. This is even more highlighted on the Figure 3 where the difference between $r_a(t)$ and $y_a(t)$ (in red on Figure 3) is always close to 0. We have the $RMSE_F = 9.5 \cdot 10^{-4}$ and $fit_F = 99\%$.

By choosing a filtered white gaussian noise as reference signal, the designed controller \mathcal{F}_1 will not only allow to follow the specific signal $r_a(t)$ but also other signals with the same frequency content and amplitude. To highlight this, we test the controller for 2 signals different from $r_a(t)$. The first one, $r_{a1}(t)$, is a sinusoid of amplitude 0.4 and of frequency 2000 Hz. The second one, $r_{a2}(t)$, is such as r_a given by (1) with $H(z)$ in (21), but generated with a different realization of the white noise $\tilde{e}(t)$. With the first signal $r_{a1}(t)$ we have the $RMSE_F = 1.2 \cdot 10^{-3}$ and $fit_F = 99\%$. With the second signal $r_{a2}(t)$ we have $RMSE_F = 1.0 \cdot 10^{-3}$ and $fit_F = 99\%$. This shows that the designed controller can be used with other reference signals.

5.2 Application of the iterative procedure

In the previous section, the reference r_a that must be tracked and that is used to identify the feedforward con-

troller \mathcal{F}_1 will generate, when applied to \mathcal{F}_1 , signals $u_a(t)$ and $y_a(t)$ that are close (in amplitude) to the ones collected in D_1 .

In this section, we will suppose that the to-be-tracked reference (1) is generated with a white noise \tilde{e} with higher variance. This reference r_b (having higher amplitude) is compared to y_1 in Figure 4. This figure shows that the amplitude of r_b is approximatively ten times higher than y_1 .

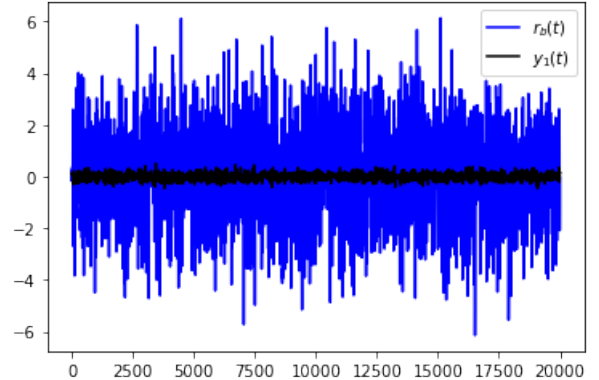


Fig. 4. New reference signal to follow r_b (in blue) compared to $y_1(t)$ used to identify the model \mathcal{M}_1 (in black)

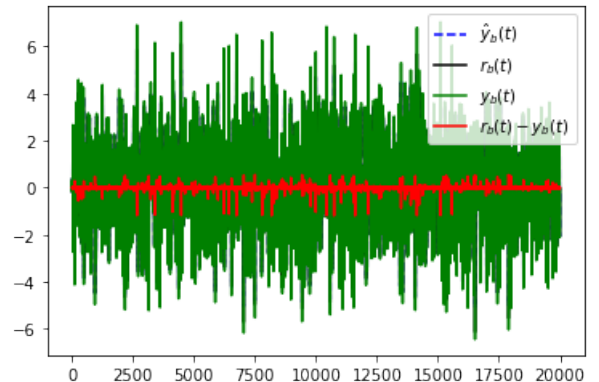


Fig. 5. Model output \hat{y}_b (blue), true system output y_b (green), reference signal r_b (black) and difference between r_b and y_b (red) when the input is the control signal $u_b(t)$ provided by the controller designed using \mathcal{M}_1 and r_b

When (11)-(12) are used to design the feedforward controller using the model identified with D_1 and with this reference r_b , the tracking performance on the true system is much poorer as indicated in Figure 5. In this case, we have $fit_F = 81\%$ and $RMSE_F = 0.31$.

Even though the tracking performance observed in figure 5 is not perfect, the application of the controller allows to obtain a certain tracking of r_b which, as indicated in Section 4, allows to collect a second dataset D_2 . This dataset is used to identify a new model \mathcal{M}_2 , using the criteria (15)-(16) and the same structure as \mathcal{M}_1 . \mathcal{M}_2 and r_b are then used to identify a new and better feedforward controller \mathcal{F}_2 , using the criterion (11)-(12). The structure of the controller \mathcal{F}_2 is the same as the one of \mathcal{F}_1 .

6. CONCLUSION

By comparing Figures 5 and 6, it is clear that the controller \mathcal{F}_2 identified with \mathcal{M}_2 and r_b achieves a much better tracking of r_b than the controller identified with \mathcal{M}_1 and r_b , as shown in Figure 6. Indeed, we obtain $fit_F = 99\%$ and a $RMSE_F = 2.7 \cdot 10^{-3}$.

The improved performance can be related to the fact that the model \mathcal{M}_2 , identified with both D_1 and D_2 , models the nonlinearity g_0 over a larger range than the model identified with only D_1 .

Note that this controller \mathcal{F}_2 also allows to efficiently track other references with similar frequency content and amplitude as r_b , as well as the initial (smaller in amplitude) reference $r_a(t)$. To highlight this, we choose as reference signal $r_c(t)$: a sinusoid of amplitude 5 and frequency 2000 Hz and feed this sinusoid to \mathcal{F}_2 .

As can be observed on figure 7, the identified controller \mathcal{F}_2 also allows to track signals with the same frequency content as $r_b(t)$. With the signal $r_c(t)$, we have a fit_F of 99% and a $RMSE_F = 1.5 \cdot 10^{-2}$. This controller also allows to track efficiently the initial reference signal $r_a(t)$. Indeed when we use the controller \mathcal{F}_2 to track $r_a(t)$, we have a fit fit_F of 99% and a $RMSE_F = 9.6 \cdot 10^{-4}$.

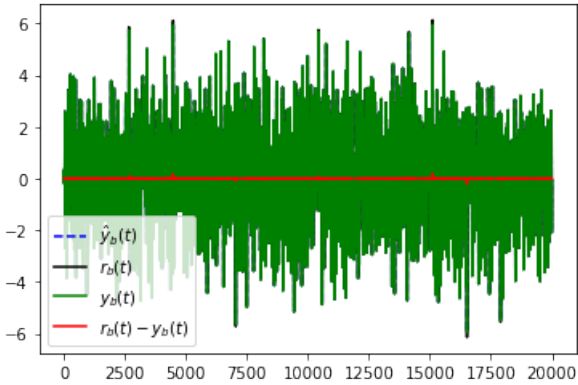


Fig. 6. Model output \hat{y}_b (blue), true system output y_b (green), reference signal r_b (black) and difference between r_b and y_b (red) when the input is the control signal $u_b(t)$ provided by the new designed controller \mathcal{F}_2

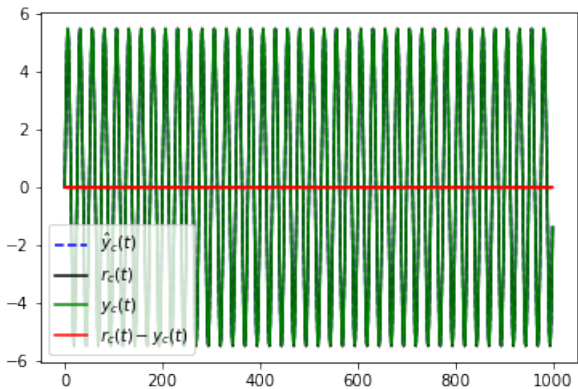


Fig. 7. Model output \hat{y}_c (blue), true system output y_c (green), reference signal r_c (black) and difference between r_c and y_c (red) when the input is the control signal $u_c(t)$ provided by the controller \mathcal{F}_2

In this paper, a model-based feedforward open-loop controller design approach for nonlinear system is presented. It leverages the use of block-oriented models to identify a model of the unknown true system and design the controller based on that model. An iterative procedure which consists of re-using the data acquired when testing controllers to re-identify a new model and then design a new controller based on it is also presented. Results on numerical applications show that the presented method is able to design feedforward controllers which allow to follow the desired reference signals. As future work, we would like to add an exploration aspect in the iterative controller design approach.

REFERENCES

- Aarnoudse, L., Ohnishi, W., Poot, M., Tacx, P., Strijbosch, N., and Oomen, T. (2021). Control-relevant neural networks for intelligent motion feedforward. In *2021 IEEE International Conference on Mechatronics (ICM)*, 1–6. IEEE.
- Bristow, D.A., Tharayil, M., and Alleyne, A.G. (2006). A survey of iterative learning control. *IEEE control systems magazine*, 26(3), 96–114.
- Dolanc, G. and Strmenik, S. (2008). Design of a nonlinear controller based on a piecewise-linear hammerstein model. In *2008 IEEE International Conference on Control Applications*, 1007–1012. IEEE.
- Firouz, Y., Goutam, S., Soult, M.C., Mohammadi, A., Van Mierlo, J., and Van den Bossche, P. (2020). Block-oriented system identification for nonlinear modeling of all-solid-state li-ion battery technology. *Journal of Energy Storage*, 28, 101184.
- Forgione, M., Bombois, X., and Van den Hof, P.M. (2015). Data-driven model improvement for model-based control. *Automatica*, 52, 118–124.
- Forgione, M. and Piga, D. (2021). dynonet: A neural network architecture for learning dynamical systems. *International Journal of Adaptive Control and Signal Processing*, 35(4), 612–626.
- Gevers, M. (2002). A decade of progress in iterative process control design: from theory to practice. *Journal of process control*, 12(4), 519–531.
- Giri, F. and Bai, E.W. (2010). *Block-oriented nonlinear system identification*, volume 1. Springer.
- Kalafatis, A.D., Wang, L., and Cluett, W.R. (2005). Linearizing feedforward-feedback control of ph processes based on the wiener model. *Journal of Process Control*, 15(1), 103–112.
- Muhammad, D., Ahmad, Z., and Aziz, N. (2021). Modeling of low density polyethylene tubular reactor using nonlinear block-oriented model. *Materials Today: Proceedings*, 42, 39–44.
- Naitali, A., Giri, F., Elayan, E., and Haloua, M. (2008). Biosocial culture inspired hierarchical algorithm for miso block oriented nonlinear system identification: application to ozone modelling. *IFAC Proceedings Volumes*, 41(2), 7433–7438.
- Poot, M., van Hulst, J., Yan, K.W., Kostić, D., Portegies, J., and Oomen, T. (2023). Feedforward control in the presence of input nonlinearities: With application to a wirebonder. *IFAC-PapersOnLine*, 56(2), 1895–1900.

- Rollins, D.K., Mei, Y., Loveland, S.D., and Bhandari, N. (2016). Block-oriented feedforward control with demonstration to nonlinear parameterized wiener modeling. *Chemical Engineering Research and Design*, 109, 397–404.
- Schoukens, M., Bai, E.W., and Rolain, Y. (2012). Identification of hammerstein-wiener systems. *IFAC Proceedings Volumes*, 45(16), 274–279.
- Schoukens, M., Marconato, A., Pintelon, R., Vandersteen, G., and Rolain, Y. (2015). Parametric identification of parallel wiener–hammerstein systems. *Automatica*, 51, 111–122.
- Van Den Hof, P.M. and Schrama, R.J. (1995). Identification and control—closed-loop issues. *Automatica*, 31(12), 1751–1770.
- Van Hulst, J., Poot, M., Kostić, D., Yan, K.W., Portegies, J., and Oomen, T. (2022). Feedforward control in the presence of input nonlinearities: a learning-based approach. *IFAC-PapersOnLine*, 55(37), 235–240.
- Vinter, R.B. and Vinter, R. (2010). *Optimal control*, volume 2. Springer.