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

A Hybrid Linear-Nonlinear ARX Model for Reliable Multi-Step Prediction : application to SwPool Benchmark

François Gauthier-Clerc[1,2,3], Hoel Le Capitaine[2,4], Fabien Claveau[2,3], Philippe Chevrel[2,3]

***Abstract*— We present a hybrid ARX model that is useful for system identification of nonlinear models. Our motivation is to combine the advantages of linear and nonlinear models in the context of extrapolation outside of the training dataset. The proposed method uses a residual hybridization approach to ensure a large linear contribution. Based on this hybrid ARX model, the proposed learning method is evaluated using the available operating data of a specific aquatic center. The results obtained on this benchmark are compared to those of traditional linear and nonlinear identification, showing that the hybrid approach achieves both the accuracy of the pure nonlinear model and the consistency of the linear ARX model. Our approach provides a promising solution for nonlinear identification, particularly for dynamical systems partially explainable by a linear model. As in the strictly linear case, the proposed model can be learned from a small volume of data, but can be enriched to improve prediction accuracy. Its potential use for data-based predictive control is particularly useful.**

I. INTRODUCTION

Climate change is a global concern that has prompted governments and industries to prioritize reducing greenhouse gas emissions. The retrofitting of existing plants with more efficient and cost-effective technologies is one way to achieve this goal. Using software retrofit is a particularly attractive approach, as it does not require significant capital investment and can provide quick returns on investment. Black box system identification and model-based control are powerful tools that can aid in this endeavor. Unlike physics-based approaches, black box models do not require expert knowledge and model adaptation based on sensors configuration. Instead, they rely on collected data to identify relationships between inputs and outputs (see [1]), making them well suited to applications where the system's physics is not always well understood or is expensive to understand.

Several applications show the relevance of model predictive control coupled with black box model to respond to our concern [1], [2]. Predictive control like EMPC (*Economical Model Predictive Control*) approach can in first place offer more energy consumption flexibility and secondly a more efficient behaviour [3]. With the ongoing transition towards renewable energy sources, the need for flexibility in industrial systems is becoming increasingly important.

[1]Purecontrol, 68 Av. Sergent Maginot, 35000 Rennes, FRANCE {firstname}.{surname}@purecontrol.com
[2]LS2N – Laboratory of digital sciences of Nantes, UMR CNRS 6004, BP 92208, 44322 Nantes
[3]IMT Atlantique, CS 20722, 44307 Nantes, FRANCE {firstname}.{surname}@imt-atlantique.fr
[4]Nantes Université, 44306 Nantes, FRANCE {firstname}.{surname}@univ-nantes.fr

Black box system identification include a lot of methods to obtain a model from data. A lot of model structures and identification processes have been proposed for both linear and non-linear contexts [4], [5]. Among the black box models, we can cite the linear polynomial models (such as ARX, OE, ARMAX), linear state space models (LSS) [4], Sparse Identification of Nonlinear Dynamics (SINDy) [6], neural networks (NN), linear parameter-varying models (LPV) [7], and many others. This field provides a complete range of model complexities that can fit any unknown model. However, the linear model is the main paradigm that has received a lot of analysis tools and theoretical results. Relying on a linear model, even if the system is non-linear, is always a good starting point and can be sufficient, especially for control.

Dealing with operating systems and using black box models without a simulator is challenging due to limited freedom in data generation and time constraints for data collection. For a cost-effective control paradigm that enables widespread deployment of such control approaches, it is crucial to first obtain a reliable and accurate model, especially in contexts characterized by small datasets and poor data coverage over the regions of interest. Linear models can offer a good trade-off in terms of data efficiency, consistency, and accuracy for certain types of systems, such as swimming pools [8], HVAC (heating, ventilation and air conditioning) systems [9], and others. Indeed, the extrapolation of linear models is powerful and predictable compared to non-linear models, and they can be characterized, which is important for control. However, linear models do not provide the best accuracy when non-linear phenomena occur, which can be detrimental to predictive control performance [1].

To the authors' knowledge, little attention has been paid to the development of methodologies for learning non-linear models that are certainly accurate, but also capable of extrapolations as good as linear models are in general. Providing a solution to this problem would allow the deployment of an efficient and low-cost nonlinear control method for all systems that can be roughly modeled by linear models. From this point, adaptive or iterative control schemes with feedback loops using new data could be used [10].

To overcome this challenge, this article proposes an identification method relying on a hybrid linear/nonlinear based ARX model. This non-linear identification scheme proposes an architecture without an implicit observer (relying on a reconstructability map [11]), emphasizing the linear contribution of this hybrid model. This approach allows a cost-effective use (no need for an expensive non-linear

observer), good extrapolation properties compared to pure non-linear methods, and competitive accuracy in the domain of the dataset. The explicability of the model is better than that of the NLARX models, by the importance given to the linear contribution. The identification scheme is well suited for predictive control, which requires precision within a bounded time horizon. We use neural networks to formulate the non-linear model as they provide good approximation performance.

As a use case, we selected the SWPool Benchmark [12] which illustrates the industrial need for identification under operational conditions that result in poor data distribution. It will make it possible to comparatively evaluate the quality of the proposed method. We rely on this dataset to draw a connection with the problem addressed in this article and to compare our approach with other system identification methods.

Hybrid linear and non-linear models are not a novel concept, as they have been explored in both time series prediction and system identification in various forms and for different purposes. In the context of time series prediction, the objective is to enhance linear expressivity with non-linear components. Zhang [13] introduced a hybrid ARX with a neural network-based noise model. Other techniques have been proposed to increase the complexity of hybrid models [14], [15]. However, these approaches are not specifically designed to handle dynamical systems identification for control.

The current state-of-the-art in system identification includes several hybridization schemes for different purposes. Among them, the generalized Non Linear State Space (g-NN-SS) model [16] model is a recent proposition in which the linear state space is coupled with a residual non-linear function. This approach has been further refined in [17], where a relevant method to initialize the learning model was proposed to enhance the optimization process. Good performances were thus obtained compared to pure non-linear methods. Other hybridization methods, closer to LPV, have also received interest [18]. To the best of our knowledge, not much effort has been devoted to leveraging this hybridization for enhancing reliability and sample efficiency in non-linear system identification. By 'reliability' of a given model, we mean its ability to predict the system's dynamics across its entire domain with precision, even beyond the training dataset.

In order to present the proposed methodology, this article will be organized as follows:

- Introduction to I/O model structure and identification method for linear and non-linear formulations.
- Introduction of the proposed hybrid method enriching NLARX formulation.
- Evaluation of the results obtained with the hybrid model and learning method proposed compared to others using SwPool benchmark [12].

II. ARX, NLARX AND NON LINEAR STATE SPACE MODELS

A. Model structure

Among the non-linear models with a reconstructability map, one of the most generic is the Deep Encoder State Space [19]. In this work, the reconstructability map function, observation function, and system function are expressed using non-linear functions in a discrete and MIMO context.

$$\begin{aligned} x[t] &= \psi_\theta(u[t:t-n_o], y[t:t-n_o]) \\ x[t+1] &= g_\theta(x[t], u[t], e[t]) \\ y[t] &= h_\theta(x[t]) + e[t] \end{aligned} \tag{1}$$

With $x[t] \in \mathbb{R}^{n_x}$, $y[t] \in \mathbb{R}^{n_y}$ and $u[t] \in \mathbb{R}^{n_u}$ denote the state, observation, and control signals, respectively. g_θ, h_θ and ψ_θ stand for the state transition function, observation function and reconstructability map respectively. θ represents the model function weights and n_o denotes the size of the delayed I/O input required to recover the state in closed form. The delayed representation $u[t+i:t]$ is equivalent to $[u[t+i]^T \cdots u[t]^T]^T$.

From this model, it is possible to derive different models such as (*non-linear output error* (NOE)) or equation error form (*non-linear auto-regressive with eXogenous* (NLARX), *non-linear auto-regressive with Moving Average and eXogenous* (NARMAX)). In this work, we concentrate on the equation error form, which enables more straightforward identification patterns.

NLARX is one of the most popular non-linear equation error models due to its training in Prediction Error Minimization (PEM), which is equivalent to a regression task. This regression task can be expressed:

$$y[t] = f_\theta(y[t-1], \cdots, y[t-n_a], u[t-1], \cdots, u[t-n_b]) + e[t] \tag{2}$$

Which admit the following state space representation:

$$\begin{aligned} x[t] &= [y[t:t-n_a], u[t-1:t-n_b]]^T \\ x[t+1] &= Px[t] + [f_\theta(x[t], u[t]), 0, \cdots]^T + Ke[t] \\ y[t] &= [1, 0, \cdots, 0]x[t] \end{aligned} \tag{3}$$

With P being a projection matrix used to shift all sub-elements of the state to comply with the definition, and $f_\theta : \mathbb{R}^{n_x \times n_u} \to \mathbb{R}^{n_y}$ representing the unfixed non-linear part of the state transition, and finally K to project the noise to the state space. From the perspective of equation 1, the reconstructability map is set to be linear and fixed as a projection matrix, as well as the observation function. The state transition function is fixed to be linear, except for the first sub-element, represented by the function f_θ. Even if we constrain the state space and significantly increase its dimension, this form can handle many systems and is easier to train with fewer parameters.

Finally, the linear model ARX can be seen as a sub-part of the NLARX, with the state transition being fully linear,

with only the first sub-element unfixed. This model is often represented using the backward shift operator q^{-1}:

$$A(q^{-1})y[t] = B(q^{-1})u[t] + e[t], \tag{4}$$

Where $A(q^{-1})$ and $B(q^{-1})$ are polynomial matrices of size $n_y \times n_y$ and $n_y \times n_u$ respectively.

This two polynomial matrices can be written as:

$$\begin{aligned} A(q^{-1}) &= I_{n_y \times n_y} + A_1 q^{-1} + A_2 q^{-2} + \cdots A_{n_a} q^{-n_a} \\ B(q^{-1}) &= B_1 q^{-1} + B_2 q^{-2} + \cdots + B_{n_b} q^{-n_b} \end{aligned} \tag{5}$$

The linear representation of the ARX state space according to equation 1 can be formulated as follows:

$$\begin{aligned} x[t] &= [y[t:t-n_a], u[t-1:t-n_b]]^T \\ x[t+1] &= Px[t] + \\ [A_1 \dots A_{n_a} B_2 \cdots B_{n_b} 0_{(n_x-1)\times n_x}]x[t] &+ [B_1\, 0 \cdots]u[t] \\ y[t] &= [1, 0, \cdots, 0]x[t] + e[t] \end{aligned} \tag{6}$$

These three models offer a good perspective from a deep black box model to a grey box model (see [5] for more examples).

B. Identification

The PEM is a commonly used method for identifying linear or non-linear I/O models. This procedure optimizes the model weights to minimize the prediction error (one step ahead) using selected data according to a scalar-value function. It is very popular since linear ARX models can be identified in closed form, and non-linear models can be optimized using traditional regression methods. The square root error for one-step prediction can be expressed as follows:

$$V_N^{(1)}(\theta, \mathcal{D}) = \sum_{t=1}^{N} ||\hat{y}[t|t-1] - y[t]||_2^2 \tag{7}$$

Here, $\hat{y}[t|t-1]$ represents the prediction given the past observation from $t-1$, θ represents the model parameters, $\mathcal{D}$ represents the dataset used for identification, and N denotes the size of that dataset. Depending on the model used, this problem can be a linear or non-linear, convex or non-convex optimization problem.

Simulation Error Minimization (SEM) is another method for identifying models, which is based not on the one step ahead prediction but on the full simulation error. This method is more suitable for output error models and can avoid some of the biases induced by the noise model in the PEM formulation [20], [21].

$$V_N^{(1:N)}(\theta, \mathcal{D}) = \sum_{t=1}^{N} ||\hat{y}[t|0] - y[t]||_2^2 \tag{8}$$

This method is unfortunately expensive and needs approximations and a dedicated optimization schema to be computed [22]. The multiple shooting [19], [23] has received recent interest due to its smoothed cost function [23] and computational efficiency. The idea is to split the simulation error into multiple bounded simulation errors using a convolution application (see equation 9). Fig. 1 illustrates an example of three bounded simulations at different starting points. The multiple shooting method proceeds like this for every data point before using all the error predictions for optimization.

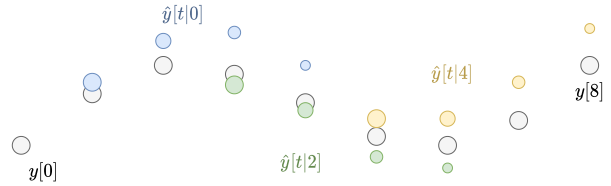

Fig. 1: Illustration of multiple shooting prediction: Three bounded simulation predictions are performed with different starting points (represented by blue, green, and yellow colors). Black points represent the original data used to estimate the model error through all those predictions.

In our context of predictive control with receding horizon, which ultimately interests us, having accurate predictions can lead to better control performance [1], [24]. Using a multi step precision criterion (bounded simulation horizon) instead of PEM or SEM is a natural adaptation to this need. The MPC Relevant Information (MPC-RI) method [25] implements this idea to identify models for predictive control purposes. The natural choice for the identification criterion can be derived from the SEM when considering a bounded horizon.

$$V_N^{(1:H)}(\theta, \mathcal{D}) = \sum_{t=H}^{N} \sum_{i=1}^{H} ||\hat{y}[t|t-i] - y[t]||_2^2 \tag{9}$$

In this formulation, where H represents the horizon size and N represents the dataset size, the expression can be viewed as the multiple shooting approximation of the SEM identification. This loss function can be efficiently optimized for both linear and non-linear models using the Back Propagation Through Time method [26] and modern automatic differentiation.

III. PROPOSED HYBRID METHOD: NLR-LARX

A. NLR-LARX Model

The proposed approach is a hybrid ARX/NLARX model that relies on residual aggregation, named NLR-LARX (NonLinearResidual-Linear-ARX). This idea is similar to Zhang's hybrid model [13] with control variables and a non-linear residual model based on the NLARX formulation. In this hybridization, the linear ARX model serves as the core predictor of the system, while the non-linear model corrects the linear model by predicting the equation error for each step of the recursive multi step prediction. The one step prediction can be formulated as follows:

$$\begin{aligned} \hat{y}[t+1|t] &= (\mathbb{I}_{n_y \times n_y} - A(q^{-1}))y[t] + B(q^{-1})u[t] + \sigma_\epsilon \tilde{e}[t] \\ \tilde{e}[t] &= f_\theta(y[t-1], \cdots, y[t-n_a], u[t-1], \cdots, u[t-n_b]) \end{aligned} \tag{10}$$

Here, $A(q^{-1})$ and $B(q^{-1})$ represent the linear ARX model, f_θ represents the non-linear function, $\tilde{e}$ represents

the non-linear equation error estimation, and σ_ϵ is a diagonal matrix. A neural network is used as the non-linear function due to its excellent approximation properties and well-established training toolbox. The constant matrix σ_ϵ ensures normalized output for the neural network during the training.

The multi step prediction is performed using the same input signals for both models. Fig. 2 represents the block diagram of this approach, where the linear noise model is applied to the neural network output. The proposed architecture enables a natural transition from a hybrid non-linear model to a classical linear model during multi step prediction, particularly when the performance of the neural network is compromised. This can be achieved in the same manner as multi step ARX prediction, by setting the future equation noise to zero. This can enable future contributions with a switch mechanism depending on the uncertainty of the neural network prediction.

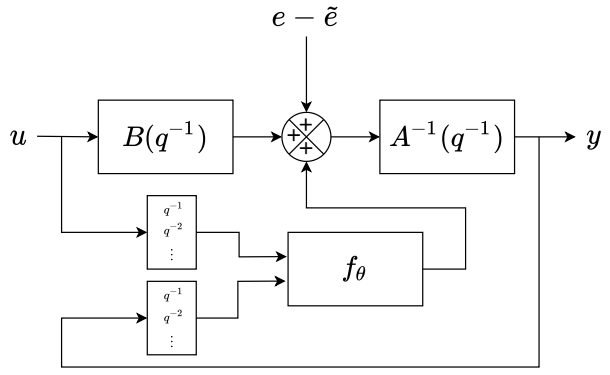

Fig. 2: Proposed hybrid NLARX scheme

The method is designed to ensure that the linear model explains the majority of the prediction, while the non-linear model corrects the linear model during recursive and long-term prediction. To achieve this, we require a linear noise model that propagates the non-linear prediction, allowing little non-linear amplitude to correct the prediction over the long run. Therefore, the non-linear model must deal with this linear closed loop during the recursive prediction. We propose a *two step identification method* with a heterogeneous identification criterion. The ARX model is trained using the PEM formulation, which allows for a closed form resolution. The error standard deviation is computed based on this first linear model. Then, the linear model is fixed and mixed with the non-linear model for non-linear training. The non-linear model is trained according to the multi step criterion. The *multi-shooting* method [19] with the appropriate prediction horizon is used to identify the residual model. This identification problem can be formulated as follows:

Step One:

$$\begin{aligned} &A^*(q^{-1}), B^*(q^{-1}) = \text{argmin} V_N^{(1)}(\{A(q^{t-1}), B(q^{-1})\}, \mathcal{D}) \\ &\text{with} : \hat{y}[t|t-1] := (I_{n_y \times n_y} - A(q^{-1}))y[t] + B(q^{-1})u[t] \\ &\qquad \sigma_\epsilon := diag(std(\hat{y}[t|t-1] - y[t])) \end{aligned} \tag{11}$$

Step Two:

$$\begin{aligned} &\theta^* = \text{argmin}_\theta V_N^{(1:H)}(\{A^*, B^*, \theta\}, \mathcal{D}) + \alpha R(\theta, \mathcal{D}) \\ &\text{with} : \hat{y}[t+k|t] = (I_{n_y \times n_y} - A^*(q^{-1}))\hat{y}[t+k-1|t] + \\ &\quad B^*(q^{-1})u[t+k-1] + \sigma_\epsilon f_\theta(\hat{y}[t+k-1|t], u[t+k-1]) \end{aligned} \tag{12}$$

Where α represents the weight of the non-linear regularization. The second part of the problem is solved using a deep learning library (Tensorflow) and automatic differentiation with the *Back Propagation Through Time* concept. The regularization term $R(\theta, \mathcal{D})$ penalizes the nonlinear prediction amplitude so that its contribution is as small as possible. The neural network is initialized with a dedicated distribution (such as GlorotUniform), but the output layer's weights are set to zero (like in [17]) in order to start the hybrid model at the ARX level.

This hybridization scheme is all the more interesting because it does not entail any model order increase compared to the original ARX model (see equation 16). The model's expressiveness remains high, as the non-linear part uses the same inputs as NLARX models. This method does not require an expensive observation system, especially with non-linear models. The ARX model is trained only for one step prediction for two reasons. Firstly, it is cost-effective and represents the expected regime of prediction throughout the horizon, thanks to the non-linear error prediction. Secondly, learning a linear model in k-step is motivated for bias correction [27], which is not necessary here since the non-linear model corrects the bias error.

This residual approach does not constrain the neural network's expressiveness and can be seen as a special case of a ResNet layer [28], employing a custom linear function rather than the identity function. Additionally, due to its use of an auto-regressive formulation, it can approximate any *recursive system* that can be represented by a continuous recursion function (see Definition 2.1 of [29]).

B. Regularization

To ensure a minor non-linear contribution, a regularization term can be added to the identification process. We propose two methods to penalize the non-linear contribution during the multi step prediction. The first method consists of using ridge regularization on the neural network output at each step. This will penalize large neural network outputs and ensure moderate non linear contribution.

$$R^{(1)}(\theta, \mathcal{D}) = \frac{1}{(N-H)H} \sum_{t=H}^{N} \sum_{k=1}^{H} ||\tilde{e}[t|t-k]||_2 \tag{13}$$

We also present a more advanced regularization approach to ensure a good extrapolation of the hybrid model. Inspired by some *Out Of Distribution (OOD) Detection* methods [30], [31], we use a grid generation of synthetic data to cover the whole range of admissible values, and penalize the neural network on this extended dataset. As explained earlier, having a non-linear prediction of zero will make the prediction identical to that of a pure linear model. Using a small weight in the regularization will not penalize the hybrid

$$X[t+1] = \begin{bmatrix} -A_1 & -A_2 & \cdots & -A_{n_a-1} & -A_{n_a} & B_2 & B_3 & \cdots & B_{n_b-1} & B_{n_b} \\ I_{n_y \times n_y} & 0 & \cdots & 0 & 0 & & & & & \\ 0 & \ddots & \ddots & \vdots & 0 & & & & & \\ \vdots & \ddots & \ddots & 0 & \vdots & & & 0 & & \\ 0 & \cdots & 0 & I_{n_y \times n_y} & 0 & & & & & \\ & & & & & 0 & \cdots & & 0 & \\ & & 0 & & & I_{n_u \times n_u} & 0 & \cdots & 0 & 0 \\ & & & & & 0 & \ddots & \ddots & 0 & \vdots \\ & & & & & \vdots & \ddots & \ddots & 0 & \\ & & & & & 0 & \cdots & 0 & I_{n_u \times n_u} & 0 \end{bmatrix} X[t] + \begin{bmatrix} B_1 \\ 0 \\ \vdots \\ 0 \\ I_{n_u \times n_u} \\ 0 \\ \vdots \\ 0 \end{bmatrix} U[t] + \begin{bmatrix} \sigma_\epsilon f_\theta(X[t], U[t]) \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

$$Y[t+1] = \begin{bmatrix} I_{n_y \times n_y} & 0 & \cdots & 0 \end{bmatrix} X[t+1], \quad \text{with} \quad X[t] = \begin{bmatrix} y[t-1]^T, \cdots, y[t-n_a]^T, u[t-1]^T, \cdots, u[t-n_b]^T \end{bmatrix}^T, Y[t] = \begin{bmatrix} y[t] \end{bmatrix} \tag{16}$$

performance too much on the training set, while penalizing the neural network in the extrapolation domain.

$$R^{(2)}(\theta, \mathcal{D}') = \frac{1}{card(\mathcal{D}')} \sum_{i=1}^{N} || f_\theta(y_i, \cdots, u_i, \cdots) ||_2 \tag{14}$$

Where $\mathcal{D}'$ represents the synthetic regularization dataset with the grid sampling and y_i, u_i element of $\mathcal{D}'$.

The main difference between the two versions is that the second version restricts the residual component to produce minimal contribution in the extrapolation domain, whereas the first version only ensures a marginal contribution of the residual model within the training dataset.

We do not expect a null contribution of the neural network across the entire system space solely through the second regularization. However, prior work [30] has shown some generalization capability of these kinds of approaches when ODD datasets are properly defined.

IV. RESULTS

A. Performance: SWPool benchmark

The SwPool Benchmark[1][12] contains a dataset from an actual aquatic center in operation. It is designed for data-driven identification of temperatures in two pools, based on a common sensor configuration and unknown system properties. The benchmark includes a training dataset, a test dataset and an extrapolation dataset. It also provides precision scores for 8-hour ahead predictions in both interpolation and extrapolation regimes. In this section, we will compare the proposed method to highlight the interest of it.

The first model is the NLARX one with a feed forward neural network in residual architecture (better accuracy than a direct feed forward model). It is formulated as equation 2 and trained with the multi step criterion. The linear ARX model is also included as a second model (equation 4) using Matlab ident. toolbox. We chose to include these two models, ARX and NLARX, in the comparison as they are parts of the proposed hybrid model. Comparing their individual performance against the performance of the entire hybrid model is of interest to us. Finally, the Deep Encoder State Space model (DESS) [19] using the Python DeepSI toolbox is also included, as it achieved one of the highest accuracies on several non-linear benchmarks [32] and represents the most general black-box state-space model with a reconstructability map. The hybrid model was trained using Matlab for the linear part and the Tensorflow library for the non-linear part. We will include the proposed method with the two proposed regularization methods separately, NLR-LARX$^{(1)}$ and NLR-LARX$^{(2)}$ for the first and second regularization method respectively (see equation 13 and 14). Between these two versions only the $R(\theta, \mathcal{D})$ regularisation function is different, in order to analyse their respective performances.

For NLR-LARX$^{(2)}$,The extrapolation data are generated using a temperature range from 26 to 31 degrees and the full range of control variables (extrapolation data have lower temperatures, around 24 degrees for both pools). All pure exogenous signals, such as outdoor temperature, are sampled from the training dataset to avoid the curse of dimensionality.

The g-NN-SS model is not included since no work has been done to apply a reconstructability map with this architecture. All non-linear training is done using the Adam [33] optimization method.

The benchmark defines the average precision through multi step prediction with following function:

$$\mathcal{L}(I, J) = \frac{1}{J + I - 1} \sum_{k=I}^{J} \sqrt{\sum_{t=H}^{N} \frac{||\hat{y}[t|t-k] - y[t]||_2^2}{N}} \tag{15}$$

Short term prediction, long term precision and average precision through the 8 hours horizon are defined with the followings quantities; $\mathcal{L}(0, H/4)$, $\mathcal{L}(3H/4, H)$ and $\mathcal{L}(0, H)$.

Hyper-parameter optimization is performed to tune all model parameters according to their specificities. Among those parameters, the horizon hyper-parameter H (which is the prediction depth criterion) is optimized only from 5 to 15 steps due to computation limitations. Besides, several empirical experiments did not show any evidence of better performance in test set using the full 48 step horizon for the training criterion.

[1]The SwPool benchmark is available at the following address `https://benchmark-datadriven-sysid.purecontrol.com`

ARX parameters :

- n_a: 3, n_b: 3
- focus: *prediction*

NLR-LARX$^{(1)}$ parameters :

- Horizon: 10
- Act. func. : *relu*
- Linear input: n_a: 3, n_b: 3
- Neural network input: n_a: 3, n_b:3
- #Layer: 1, #neurons: 32
- Learning rate: 3.10^{-4}
- Batch size: 64
- #Epochs: 300
- Alpha : 0.0001

NLARX parameters:

- Horizon : 15
- Act. func. : *relu*
- n_a: 3, n_b: 3
- #layers: 2, #neurons: 16
- Learning rate: 10^{-4}
- Batch size: 128
- #epochs: 300
- L_2 Norm Pem. : 10^{-4}

DESS parameters:

- Horizon : 15
- n_x: 6, n_o: 18
- Act. func.: *tanh*,
- #layers: 2, #neurons: 16
- Learning rate : 10^{-3}
- Batch size: 256,
- #Epochs: 400

TABLE I: All hyper-parameters used in the numerical comparison.

All optimized hyper-parameters are listed TABLE. I. They are obtained with a random search with around 300 trials per model. The optimal hyper-parameters for NLR-LARX$^{(2)}$ are identical to those for NLR-LARX$^{(1)}$(except for n_a=1 and α=0.001), and are not given twice in TABLE I.

The benchmark results are presented in Table II. With the exception of the ARX model (which is fitted with a deterministic algorithm), all results are computed 10 times to estimate the mean and standard deviation that are reported in the table.

The NLR-LARX$^{(1)}$ model provides the best accuracy on the test set compared to the two original methods (ARX and NLARX). Overall, all non-linear models trained using the same multiple shooting method provide similar performance on the test set. NLR-LARX$^{(1)}$ is slightly better due to reduced variance in the model performance. NLR-LARX$^{(2)}$ is worst in terms of test accuracy (around 0.16 compared to 0.15 for other non-linear models; see the first column of TABLE II), indicating some negative effects of the proposed regularization in the interpolation context.

As expected, the linear model provides the best extrapolation performance across all four extrapolation scenarios of the benchmark. Both non-linear models (NLARX & DESS) produce very poor accuracy in all scenarios. This highlights the limitations of pure non-linear model for consistent black box modeling.

The hybrid model with the first regularization does not provide better extrapolation than other non-linear models, which suggests that the residual non-linear model produces unpredictable contribution. On the other hand, the second regularization method, NLR-LARX$^{(2)}$, which penalizes the residual model output in some synthetic extrapolation data, performs as expected with accuracy close to the linear model for several scenarios. This result tends to show the ability of the proposed method to merge the best of the both worlds, non-linear interpolation, and linear extrapolation (or at least

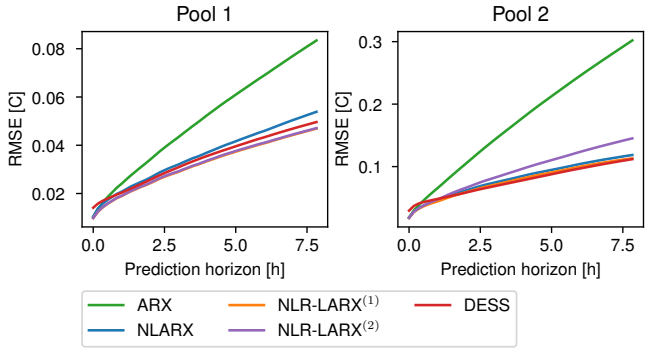

Fig. 3: Five 48 steps prediction on test data.

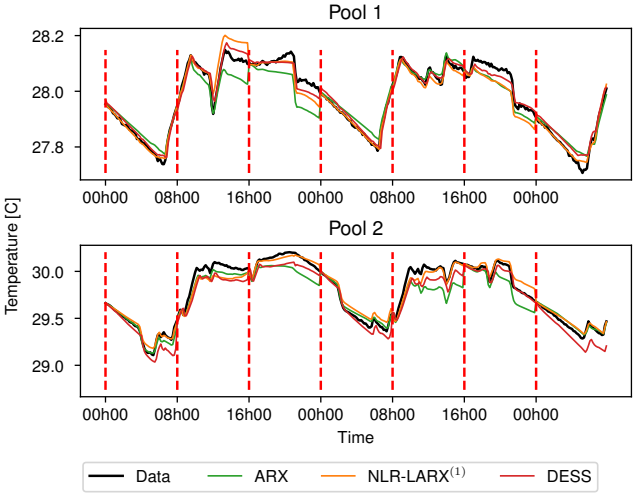

Fig. 4: Prediction example for several prediction horizons.

be able to move in the Pareto front). Some work needs to be done to improve the regularization to achieve the best non-linear interpolation while remaining close to the linear extrapolation capability.

The precision of both pool signals according to the prediction depth and the model is presented in Fig. 3 (test set only). As expected, the linear model performs the worst, especially with the second pool. All non-linear models produce less error, notably in the long term prediction. The DESS produces very competitive accuracy but performs worse in the short term due to inaccurate state space estimation (as noted in the original paper [19]).

An example of predictions for different horizons is shown in Fig. 4. At each vertical red line, all models are initialized with the real observations according to their specifications.

B. NLR-LARX behaviour

By examining the dynamic relationship between the linear and non-linear components, we can gain further insight into the performance of our model. Specifically, it is crucial to ensure that the non-linear contribution remains marginal compared to the linear contribution.

We draw our attention to this behavior by comparing the variance of the residual nonlinear output, while taking into account its normalization matrix σ_ϵ. In order to estimate the magnitude of the variance, we propose including the variance of the discrete derivative of the linear ARX model alone

	$\mathcal{L}(1,H)$	$\mathcal{L}(1,H/4)$	$\mathcal{L}(3H/4,H)$	S1 $\mathcal{L}(1,H)$	S2 $\mathcal{L}(1,H)$	S3 $\mathcal{L}(1,H)$	S4 $\mathcal{L}(1,H)$
ARX	0.293±0.00	0.113±0.00	0.463±0.00	0.276±0.00	**0.284±0.00**	**0.347±0.00**	**0.327±0.00**
NLARX	0.159±0.0073	0.086±0.0013	0.221±0.0139	0.303±0.0798	0.669± 0.1556	1.524±0.7346	0.928±0.3187
DESS	0.151±0.0102	0.090±0.0038	0.207±0.0180	0.565±0.2060	1.162±0.4327	3.348±1.7979	1.274±0.6707
NLR-LARX$^{(1)}$	**0.147±0.0033**	**0.081±0.0008**	**0.204±0.0053**	0.332±0.0703	0.697±0.2730	3.076±1.1952	1.543±0.3662
NLR-LARX$^{(2)}$	0.164±0.0091	0.085±0.0016	0.238±0.0170	**0.221±0.0236**	0.431±0.1028	0.486±0.0777	0.404±0.0543

TABLE II: Swp Benchmark score:
$\mathcal{L}(1,H)$,$\mathcal{L}(1,H/4)$,$\mathcal{L}(1,3H/4)$ are scores for the test set, and Si represents the score for the ith extrapolation scenario.

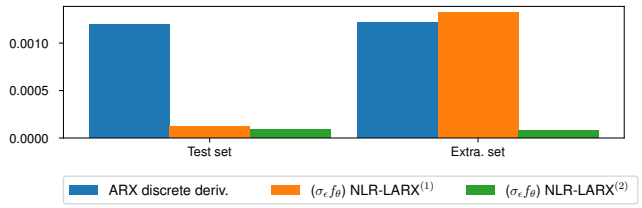

Fig. 5: Variance of the discrete derivative of the linear ARX model and the non-linear residuals of both NLR-LARX$^{(1)}$ and NLR-LARX$^{(2)}$ models for the test set and extrapolation set.

(which can be computed after hand with $\hat{y}[t+1|t-k] - \hat{y}[t|t-k]$). By considering only the discrete derivative, we can make a more accurate comparison, as the residual portion is not necessary to predict the entire transition function.

This comparison is shown in Fig. 5 for both the test set and the extrapolation dataset, using both the NLR-LARX$^{(1)}$ and NLR-LARX$^{(2)}$ models, as well as the linear model. All variances were estimated through 8-hour recursive prediction for the entire dataset, using the same 10 models used in the benchmark score. Consistent with our expectations, the non-linear residual variance is relatively small compared to the variance of the linear discrete derivative. We observe a slightly more variance with the NLR-LARX$^{(1)}$ model compare to NLR-LARX$^{(2)}$, which can explain the better performance. However, in the extrapolation data, the NLR-LARX$^{(1)}$ model exhibits a significant increase in output variance compared to its interpolation regime, with 10 times more variance. On the other hand, the NLR-LARX$^{(2)}$ model with the second regularization allows for a relatively constant variance of the non-linear residual model. This result is expected since the $R^{(1)}$ regularization does not constrain the residual model in the extrapolation regime, unlike $R^{(2)}$. While this stable contribution helps with extrapolation (as shown with $R^{(2)}$), achieving linear equivalent extrapolation accuracy would require the residual output to have a zero or near zero amplitude (not exactly the case with $R^{(2)}$).

In addition to the amplitude comparison, we can compute the cross-correlation between the linear discrete derivative of the ARX model and the non-linear output of the NLR-LARX$^{(1)}$ model.

Pool 1

	DD ARX	$(\sigma_\epsilon f_\theta)$ NLR-LARX$^{(1)}$
DD ARX	1.0	-0.063
$(\sigma_\epsilon f_\theta)$ NLR-LARX$^{(1)}$	-0.063	1.0

Pool 2

	DD ARX	$(\sigma_\epsilon f_\theta)$ NLR-LARX$^{(1)}$
DD ARX	1.0	-0.027
$(\sigma_\epsilon f_\theta)$ NLR-LARX$^{(1)}$	-0.027	1.0

TABLE III: Covariance matrices between the residual non-linear model of NLR-LARX$^{(1)}$ and the Discrete Derivative of the ARX (DD ARX) model in the test set.

The ARX output and residual model exhibit no linear correlation. As described in Zhang's hybrid ARX model [13], the ARX component captures all of the linear correlations between the lagged observations and the output predictions. This constraint forces the neural network to model only the non-linear relationships between the input and output variables.

To summarize this empirical study, the proposed structure exhibits good characteristics in terms of both prediction accuracy in the interpolation regime and prediction ability in the extrapolation regime. However, there are limits to achieving excellent simultaneous interpolation and extrapolation performance.

V. CONCLUSIONS

In this paper, a hybrid linear-nonlinear autoregressive black box model has been proposed, as a useful support for system identification. It is particularly suited to the representation of physical processes whose behavior can be largely explained by a linear model. The identification method that we have recommended and implemented proceeds to a residual hybridization. It combines a state-of-the-art ARX model with a trained neural network in a multi step error minimization process. In order to guarantee the predominance of the linear part of the model, we have developed a two-step learning process. The residual hybridization technique ensures consistent use of the linear model with neural network prediction across multi step predictions. Two regularizations have been proposed to limit the non-linear contribution in both the extrapolation and interpolation regimes.

The results obtained from the numerical experiments on the SwPool benchmark show that the learned hybrid model reaches on the test set (interpolation) an accuracy equivalent to that obtained with a purely non-linear model. At the same time, it offers much higher performance in extrapolation, with an accuracy close to that obtained by an optimized linear model alone. The empirical analysis of the output of the neural network confirms a weak contribution for the prediction, compared to that of the linear part. This allows a linear interpretation/explainability of the black box model [2]. Regularizations considered have shown their effectiveness. We will continue our search however, to have a regularization

[2]the best of both worlds?

requiring no compromise. Future work will also focus on exploring the benefits of the model we proposed, to be used for predictive control applications.

NOMENCLATURE

PEM	Prediction Error Minimization
SEM	Simulation Error Minimization
MPC-RI	Model Predictive Control Relevant Information
EMPC	Economical Model Predictive Control
LSS	Linear State Space
ARX	Auto-Regressive with eXogenous
ARMAX	Auto-Regressive Moving Average with eXogenous
OE	Ouput Error model
LPV	Linear Parameter Varying
NLARX	Non-Linear Auto-Regressive with eXogenous
NOE	Non-linear Output Error
DESS	Deep Encoder State Space
g-NN-SS	generalized Non Linear State Space
NLR-LARX	Non-Linear-Residual Linear Auto-Regressive with eXogenous
NN	Neural Network
OOD	Out Of Distribution
DD ARX	Discrete Derivative of Auto-Regressive with eXogenous

REFERENCES

[1] P. C. Blaud, P. Chevrel, F. Claveau, P. Haurant, and A. Mouraud, "From multi-physics models to neural network for predictive control synthesis," *Optimal Control Applications and Methods*, vol. 44, no. 3, pp. 1394–1411, 2023.

[2] M. Wallace, R. McBride, S. Aumi, P. Mhaskar, J. House, and T. Salsbury, "Energy efficient model predictive building temperature control," *Chemical Engineering Science*, vol. 69, no. 1, pp. 45–58, 2012.

[3] J. P. D. Marín, F. V. García, and J. R. G. Cascales, "Use of a predictive control to improve the energy efficiency in indoor swimming pools using solar thermal energy," *Solar Energy*, vol. 179, pp. 380–390, 2019.

[4] L. Ljung, *System Identification: Theory for the User*. Pearson Education, 1998.

[5] J. Schoukens and L. Ljung, "Nonlinear system identification: A user-oriented road map," *IEEE Control Systems Magazine*, vol. 39, no. 6, pp. 28–99, 2019.

[6] S. L. Brunton, J. L. Proctor, and J. N. Kutz, "Discovering governing equations from data by sparse identification of nonlinear dynamical systems," *Proceedings of the National Academy of Sciences*, vol. 113, no. 15, pp. 3932–3937, 2016.

[7] C. A. Thilker, P. Bacher, D. Cali, and H. Madsen, "Identification of non-linear autoregressive models with exogenous inputs for room air temperature modelling," *Energy and AI*, vol. 9, p. 100165, 8 2022.

[8] Y. Dong, H. Yonghong, and X. Gaohong, "Design of indoor swimming pool water temperature control system based on fuzzy controller and smith predictor," in *Proceedings of 2011 International Conference on Electronic & Mechanical Engineering and Information Technology*, vol. 9, 2011, pp. 4678–4681.

[9] M. Vašak, A. Starčić, and A. Martinčević, "Model predictive control of heating and cooling in a family house," in *2011 Proceedings of the 34th International Convention MIPRO*, 2011, pp. 739–743.

[10] Y. Zhu, R. Patwardhan, S. B. Wagner, and J. Zhao, "Toward a low cost and high performance mpc: The role of system identification," *Computers & Chemical Engineering*, vol. 51, pp. 124–135, 2013, cPC VIII.

[11] M. Forgione, M. Mejari, and D. Piga, "Learning neural state-space models: do we need a state estimator?" 2022.

[12] F. Gauthier-Clerc, H. Le Capitaine, F. Claveau, and P. Chevrel, "Operating data of a specific aquatic center as a benchmark for dynamic model learning: search for a valid prediction model over an 8-hour horizon," in *arXiv/2303.07195*, 2023.

[13] G. Zhang, "Time series forecasting using a hybrid arima and neural network model," *Neurocomputing*, vol. 50, pp. 159–175, 2003.

[14] M. Khashei and M. Bijari, "A novel hybridization of artificial neural networks and arima models for time series forecasting," *Applied Soft Computing*, vol. 11, no. 2, pp. 2664–2675, 2011, the Impact of Soft Computing for the Progress of Artificial Intelligence.

[15] E. Dave, A. Leonardo, M. Jeanice, and N. Hanafiah, "Forecasting indonesia exports using a hybrid model arima-lstm," *Procedia Computer Science*, vol. 179, pp. 480–487, 2021, 5th International Conference on Computer Science and Computational Intelligence 2020.

[16] M. Forgione and D. Piga, "Model structures and fitting criteria for system identification with neural networks," in *2020 IEEE 14th International Conference on Application of Information and Communication Technologies (AICT)*, 2020, pp. 1–6.

[17] M. Schoukens, "Improved initialization of state-space artificial neural networks," in *2021 European Control Conference (ECC)*. IEEE, 2021, pp. 1913–1918.

[18] R. Magalhães, C. Fontes, L. Almeida, and M. Embiruçu, "Identification of hybrid arx–neural network models for three-dimensional simulation of a vibroacoustic system," *Journal of Sound and Vibration*, vol. 330, no. 21, pp. 5138–5150, 2011.

[19] G. Beintema, R. Toth, and M. Schoukens, "Nonlinear state-space identification using deep encoder networks," in *Proceedings of the 3rd Conference on Learning for Dynamics and Control*, ser. Proceedings of Machine Learning Research, A. Jadbabaie, J. Lygeros, G. J. Pappas, P. A. Parrilo, B. Recht, C. J. Tomlin, and M. N. Zeilinger, Eds., vol. 144. PMLR, 07 – 08 June 2021, pp. 241–250.

[20] L. Ljung, *System Identification: Theory for the User*, ser. Prentice Hall information and system sciences series. Prentice Hall PTR, 1999.

[21] M. Farina and L. Piroddi, "Some convergence properties of multi-step prediction error identification criteria," in *2008 47th IEEE Conference on Decision and Control*, 2008, pp. 756–761.

[22] F. Ding, P. X. Liu, and G. Liu, "Gradient based and least-squares based iterative identification methods for oe and oema systems," *Digital Signal Processing*, vol. 20, no. 3, pp. 664–677, 2010.

[23] A. H. Ribeiro, K. Tiels, J. Umenberger, T. B. Schön, and L. A. Aguirre, "On the smoothness of nonlinear system identification," *Automatica*, vol. 121, p. 109158, nov 2020.

[24] D. I. Mendoza-Serrano and D. J. Chmielewski, "Smart grid coordination in building hvac systems: Empc and the impact of forecasting," *Journal of Process Control*, vol. 24, no. 8, pp. 1301–1310, 2014, economic nonlinear model predictive control.

[25] R. Gopaluni, R. Patwardhan, and S. Shah, "Mpc relevant identification—tuning the noise model," *Journal of Process Control*, vol. 14, no. 6, pp. 699–714, 2004.

[26] P. Werbos, "Backpropagation through time: what it does and how to do it," *Proceedings of the IEEE*, vol. 78, no. 10, pp. 1550–1560, 1990.

[27] J. Zhao, Y. Zhu, and R. Patwardhan, "Identification of k-step-ahead prediction error model and mpc control," *Journal of Process Control*, vol. 24, pp. 48–56, 2014.

[28] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[29] J. HAMMER, "Non-linear systems: stability and rationality," *International Journal of Control*, vol. 40, no. 1, pp. 1–35, 1984.

[30] D. Hendrycks, M. Mazeika, and T. Dietterich, "Deep anomaly detection with outlier exposure," 2019.

[31] J. Chen, Y. Li, X. Wu, Y. Liang, and S. Jha, "Robust out-of-distribution detection for neural networks," 2021.

[32] A. Marconato, J. Sjöberg, J. Suykens, and J. Schoukens, "Identification of the silverbox benchmark using nonlinear state-space models," *IFAC Proceedings Volumes*, vol. 45, no. 16, pp. 632–637, 2012, 16th IFAC Symposium on System Identification.

[33] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014.