Nonlinear System Identification using Deep Neural Network with Kernel-Based Regularization

Tatsuya Okuzawa[1†], Kentaro Mitsuma[1], Takahiro Kawaguchi, and Seiji Hashimoto[1]

[1]Graduate School of Science and Technology, Gunma University, Gunma, Japan
(Tel: +81-277-30-1754, 1706, 1741; E-mail: t231d019, t190d106, kawaguchi, hashimotos@gunma-u.ac.jp)

Abstract: This paper addresses a nonlinear system identification problem and proposes a model that combines neural networks with FIR (Finite Impulse Response) model. By highlighting the similarity between FIR model and temporal neural networks, the combined model can be regarded as a single DNN (deep neural network). This interpretation enables efficient training using deep learning frameworks. Furthermore, the paper introduces the application of Kernel-based regularization, a technique developed in the linear system identification field, to the FIR part of the proposed model. This method enhances model accuracy by leveraging prior knowledge that the impulse response of a stable linear system decays exponentially. The effectiveness of the proposed approach is validated through numerical simulations.

Keywords: Kernel regularization, Deep neural network, Nonlinear system identification

1. INTRODUCTION

System identification is a modeling method used to construct a mathematical model using input-output data from a target system, with numerous methods proposed so far. Regarding the identification of linear systems, approaches utilizing ARX (Auto Regressive eXogenous) model and FIR (Finite Impulse Response) model are well-known [1]. In recent years, identification methods for FIR model based on kernel regularization have been attracting attention [2, 3]. In this method, the identification accuracy is improved by utilizing prior information that the impulse responses of stable linear systems decay exponentially.

Compared to traditional linear system identification theory, methodologies for nonlinear system identification remain in active development. Recently, there has been growing interest in control system design and analysis theories rooted in nonlinear models [4], driven by the practical necessity of achieving robust control performance even in the presence of intricate nonlinear systems. Specifically, there have been investigations into methodologies incorporating DNN (Deep Neural Networks) [5]. Conversely, models based on deep learning exhibit complexity and limited interpretability, prompting the utilization of block-oriented models like the Hammerstein-Wiener model in certain cases [4, 6].

This paper considers a method for constructing a block-oriented model by combining FIR models with deep learning models. By noting that the FIR model is a special case of TCN (Temporal Convolutional Network) [7], it can be interpreted that a block-oriented model containing FIR models constitutes “a single DNN”. The paper proposes incorporating regularization methods based on kernels developed in linear system identification into these deep learning models.

The effectiveness of the proposed method is demonstrated through numerical examples.

† Tatsuya Okuzawa is the presenter of this paper.

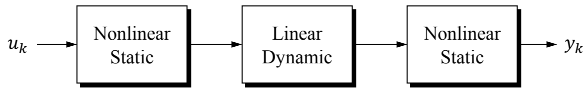


Fig. 1: Hammerstein-Wiener Model

2. PROBLEM SETTING

Suppose that input-output data $\{u_1, \ldots, u_N\}$ and $\{y_1, \ldots, y_N\}$ from a nonlinear system with a length of N is obtained. The problem considered in this paper is to construct a model that accurately represents the input-output relationship using this input-output data. Specifically, the objective is to construct a simulation model driven solely by the input, given by

$$\hat{y}(k) = f(\mathcal{U}_k), \tag{1}$$

where $\mathcal{U}_k = \{u_\kappa \mid \kappa \leq k\}$, and $\hat{y}_k$ denotes the model output.

3. MODEL STRUCTURE

This paper considers constructing a block-oriented model for nonlinear system identification. As an example of the block-oriented models, the Hammerstein-Wiener model is shown in Fig. 1. This model represents a nonlinear system by connecting a linear dynamic element and nonlinear static elements. This paper considers using the FIR model as the linear dynamic element and three-layer neural networks (NN: Neural Networks) as the nonlinear static elements in such block-oriented models.

3.1. FIR model

The model that expresses the relationship between the model input u_k and the output $\hat{y}_k$ as

$$\hat{y}_k = g_0 u_k + g_1 u_{k-1} + \cdots + g_n u_{k-n} \tag{2}$$

is called the FIR model, where $g_0, g_1, \ldots, g_n$ represent the finite impulse response of the system. Extending this

Manuscript 257 submitted to 2024 SICE Festival 2024 with Annual Conference.
Received April 22, 2024.

model to a multiple-input multiple-output between the input $\boldsymbol{u}_k \in \mathbb{R}^m$ and the output $\hat{\boldsymbol{y}}_k \in \mathbb{R}^p$,

$$\hat{\boldsymbol{y}}_k = \boldsymbol{G}_0\boldsymbol{u}_k + \boldsymbol{G}_1\boldsymbol{u}_{k-1} + \cdots + \boldsymbol{G}_n\boldsymbol{u}_{k-n} \tag{3}$$

is obtained, where n denotes the model order, and $\boldsymbol{G}_i \in \mathbb{R}^{p\times m}$, $i = 0, 1, \ldots, n$, are the finite impulse response matrices. Since this model corresponds to the convolution in the temporal domain between the finite impulse response matrices $\boldsymbol{G}_i$ and the input $\boldsymbol{u}_k$, it is a special case of TCN (Temporal Convolutional Network). Therefore, the model is described as

$$\begin{aligned}\mathrm{TCN}(\boldsymbol{u}_{k-n:k} \mid \mathcal{G}) = \boldsymbol{G}_0\boldsymbol{u}_k + \boldsymbol{G}_1\boldsymbol{u}_{k-1} + \\ \cdots + \boldsymbol{G}_n\boldsymbol{u}_{k-n},\end{aligned} \tag{4}$$

where $\boldsymbol{u}_{k-n:k} = \{\boldsymbol{u}_\kappa \mid k-n \le \kappa \le k\}$ represents a sequence of input vectors from $k-n$ to k, and $\mathcal{G} = \{\boldsymbol{G}_0, \ldots, \boldsymbol{G}_n\}$ represents the set of finite impulse response matrices.

3.2. Three-layer NN model

As a nonlinear static system, a three-layer neural network defined by

$$\mathrm{NN}(\boldsymbol{x} \mid \mathcal{P}) = \mathrm{FC}\left(\mathrm{ReLU}\big(\mathrm{FC}(\boldsymbol{x} \mid \boldsymbol{W}_1, \boldsymbol{b}_1)\big) \mid \boldsymbol{W}_2, \boldsymbol{b}_2\right) \tag{5}$$

is utilized, where $\mathrm{FC}(\cdot \mid \cdot, \cdot)$ is a fully connected (FC) layer [8] defined with vector $\boldsymbol{x}$ and appropriate sized $\boldsymbol{W}$ and $\boldsymbol{b}$ as

$$\mathrm{FC}(\boldsymbol{x} \mid \boldsymbol{W}, \boldsymbol{b}) = \boldsymbol{W}\boldsymbol{x} + \boldsymbol{b}. \tag{6}$$

The matrices $\boldsymbol{W}_1$, $\boldsymbol{b}_1$, $\boldsymbol{W}_2$, and $\boldsymbol{b}_2$ are the weights and biases of the first FC layer and the second FC layer, respectively. The ReLU (Rectified Linear Unit) is defined as

$$\mathrm{ReLU}(\boldsymbol{x}) = \max(\boldsymbol{x}, 0), \tag{7}$$

where the $\max$ function in (7) is applied for each element of $\boldsymbol{x}$. Furthermore, $\mathcal{P}$ is a set of parameters defined as

$$\mathcal{P} = \{\boldsymbol{W}_1, \boldsymbol{b}_1, \boldsymbol{W}_2, \boldsymbol{b}_2\}. \tag{8}$$

In the FC layer in (6), the dimension of the output is referred to as the number of nodes.

3.3. Block-oriented model

Using the FIR model and 3-layer NN as elements of the block oriented model, the model in Fig. 1 can be written as

$$\hat{y}_k = \mathrm{NN}(\boldsymbol{z}_k^{(1)}, \mathcal{P}^{(1)}), \tag{9}$$
$$\boldsymbol{z}_k^{(1)} = \mathrm{TCN}(\boldsymbol{z}_{k-n:k}^{(2)} \mid \mathcal{G}^{(2)}), \tag{10}$$
$$\boldsymbol{z}_k^{(2)} = \mathrm{NN}(u_k, \mid \mathcal{P}^{(3)}), \tag{11}$$

where the parameters in this model are $\mathcal{P}^{(1)}$, $\mathcal{G}^{(2)}$, and $\mathcal{P}^{(3)}$. As a further generalization, a deeper model as

$$\hat{y}_k = \mathrm{NN}(\boldsymbol{z}_k^{(1)}, \mathcal{P}^{(1)}), \tag{12}$$
$$\boldsymbol{z}_k^{(1)} = \mathrm{TCN}(\boldsymbol{z}_{k-n:k}^{(2)} \mid \mathcal{G}^{(2)}), \tag{13}$$
$$\vdots$$
$$\boldsymbol{z}_k^{(M-1)} = \mathrm{NN}(u_k, \mid \mathcal{P}^{(M)}) \tag{14}$$

can be considered, where M is the total number of 3-layer NNs and TCNs to be used. These models can be considered as a single DNN since both NN and TCN are members of the neural network family. Therefore, one can efficiently learn the parameters using deep learning frameworks such as PyTorch [9] or the Deep Learning Toolbox in MATLAB.

4. LEARNING METHOD USING KERNEL-BASED REGULARIZATION

4.1. Kernel-based regularization

In the FIR model in (2), $g_0, \ldots, g_n$ represent the finite impulse response of the system, and they decay exponentially for the stable system. To capture this feature, an estimation method for the FIR parameters has been proposed to minimize the cost function

$$J(\boldsymbol{g}) = \sum_{k=1}^{N}(y_k - \hat{y}_k)^2 + \lambda \boldsymbol{g}^\top \boldsymbol{K}^{-1}\boldsymbol{g}, \tag{15}$$

where $\boldsymbol{g} = [g_0, \ldots, g_n]^\top$. The matrix $\boldsymbol{K}$ is called the Gram matrix, where its (i, j) element is determined by the kernel function $k(i, j)$, which is designed to leverage the information that the impulse response decays exponentially. One of the most common kernels is the DC (Diagonal-Correlated) kernel, defined as

$$k(i, j) = \beta\alpha^{\frac{i+j}{2}}\rho^{|i-j|}. \tag{16}$$

The hyperparameters $0 < \alpha < 1$, $\beta > 0$, and $0 < \rho < 1$ are determined according to the prior information of the system.

4.2. Learning method of deep learning models with Kernel-based regularization

As mentioned earlier, TCN can be considered as a generalization of FIR. From this perspective, applying kernel-based regularization to the TCN parameters $\mathcal{G}^{(i)}$ of the model described in the previous section is expected to enable the construction of a model with exponential decay characteristics. Specifically, the parameters should be trained with the cost function

$$J(\boldsymbol{\Phi}) = \sum_{k=1}^{N}(y_k - \hat{y}_k)^2 + \lambda \sum_{i\in\mathcal{T}} R(\mathcal{G}^{(i)} \mid \boldsymbol{K}), \tag{17}$$

where

$$R(\mathcal{G} \mid \boldsymbol{K}) = \sum_{i=1}^{p}\sum_{j=1}^{m} \boldsymbol{g}_{ij}^\top \boldsymbol{K}^{-1}\boldsymbol{g}_{ij}, \tag{18}$$

and $\boldsymbol{\Phi} = \{\mathcal{P}^{(1)}, \mathcal{G}^{(2)}, \dots, \mathcal{P}^{(M)}\}$, and $\mathcal{T}$ denotes the set of indices corresponding to TCNs. The vector $\boldsymbol{g}_{ij}$ is defined as

$$\boldsymbol{g}_{ij} = \begin{bmatrix} G_0(i,j) & G_1(i,j) & \dots & G_n(i,j) \end{bmatrix}, \tag{19}$$

which represents the impulse response from the j-th input to the i-th output.

In practical applications, directly minimizing the evaluation function (17) is not desirable due to the numerical instability of computing the inverse matrix $\boldsymbol{K}^{-1}$ of the Gram matrix $\boldsymbol{K}$. In linear system identification with kernel-based regularization, it is known that the parameters $\boldsymbol{g}$ that minimize the evaluation function (15) satisfy

$$\boldsymbol{g} = \boldsymbol{K}\boldsymbol{\theta}, \tag{20}$$

where $\boldsymbol{\theta}$ is a vector of the same size as $\boldsymbol{g}$. Following this, the equality

$$\boldsymbol{g}_{ij} = \boldsymbol{K}\boldsymbol{\theta}_{ij} \tag{21}$$

is assumed in the proposed method as well. This rewrites the regularization term as

$$R(\boldsymbol{\Theta} \mid \boldsymbol{K}) = \sum_{i=1}^{p}\sum_{j=1}^{m} \boldsymbol{\theta}_{ij}^{\top}\boldsymbol{K}\boldsymbol{\theta}_{ij}, \tag{22}$$

where $\boldsymbol{\Theta} = \{\boldsymbol{\theta}_{ij} \mid i = 1, \dots, p, j = 1, \dots, m\}$. Note that this expression avoids the calculation of $\boldsymbol{K}^{-1}$.

Since TCN and its corresponding gradient calculation are pre-implemented in deep learning frameworks, it is desirable for the implementation of the training process to be consistent with them. Therefore, minimizing the evaluation function

$$J(\boldsymbol{\Phi}, \boldsymbol{\Theta}) = \sum_{k=1}^{N}(y_k - \hat{y}_k)^2 + \sum_{i \in \mathcal{T}} \left\{ \lambda R(\boldsymbol{\Theta}^{(i)} \mid \boldsymbol{K}) + \gamma E(\mathcal{G}^{(i)}, \boldsymbol{\Theta}^{(i)}) \right\} \tag{23}$$

is proposed. Note that the error term

$$E(\mathcal{G}, \boldsymbol{\Theta}) = \sum_{i=1}^{p}\sum_{j=1}^{m}(\boldsymbol{g}_{ij} - \boldsymbol{K}\boldsymbol{\theta}_{ij})^2 \tag{24}$$

is added instead of solving the constrained optimization problem with (21).

5. NUMERICAL EXAMPLE

In this section, the effectiveness of the proposed method is verified through numerical examples. The system to be identified is shown in Fig. 2, which is a single-input single-output system. The filter $G_1(s)$ is designed as a 3rd order Chebyshev filter with cutoff frequency of 4.4 kHz and a passband ripple of 0.5 dB, and $G_2(s)$ is designed as a 3rd order inverse Chebyshev filter with stopband attenuation of 40 dB starting at 5 kHz. The nonlinear static elements $f[\cdot]$ is a Leaky-ReLU-like function

$$f(x) = \begin{cases} 0.5x & x \geq 0 \\ x & x < 0 \end{cases}. \tag{25}$$

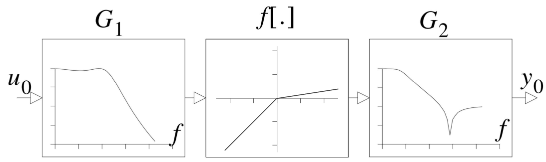


Fig. 2: Target System

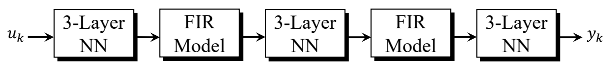


Fig. 3: Proposed Model

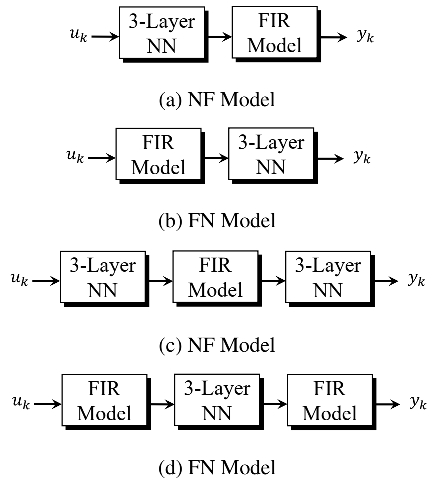
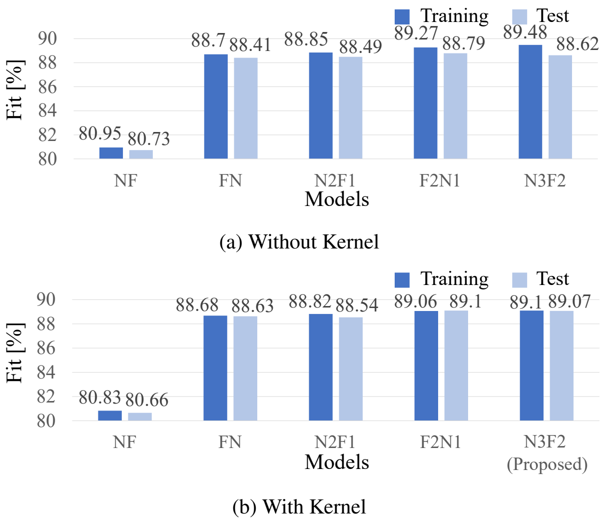


Fig. 4: Smaller Models

The system is driven by a zero-mean Gaussian white input, and the output is obtained with a sampling frequency of $51.2\,\mathrm{kHz}$. A zero-mean Gaussian white noise is added to the output, with a signal-to-noise ratio (SNR) of $20\,\mathrm{dB}$. Two input-output datasets are obtained for training and testing purposes, with the proposed method applied to the training data.

The proposed model is illustrated in Fig. 3 and is referred to as the N3F2 model in this paper. The hyperparameters are set as follows: the order and the number of outputs of all the FIR models are $n = 100$, $p = 5$, respectively, and the number of nodes in the 3-layer neural networks are set to 100 and 5, respectively. The number of outputs of the last 3-layer neural network is set to 1 to match the target system. Furthermore, DC kernel-based regularization is applied for the FIR model parts. The kernel hyperparameters are set as $\alpha = 0.9$, $\beta = 100$, $\rho = 0.9$, $\lambda = 1$, and $\gamma = 100$. To clarify the usefulness of the proposed model, smaller models shown in Fig. 4 are also constructed. The models are referred to as the NF model, FN model, F2N1 model, N2F1 model, and F2N1 model respectively. The hyperparameters for each model are set the same as the proposed model.

The accuracy evaluation results for each model are shown in Fig. **??** (**??**) and (**??**), illustrating the results without and with kernel regularization, respectively. From the figures, it can be observed that the F2N1 and N3F2 models demonstrate higher accuracy compared to



(a) Without Kernel

(b) With Kernel

Fig. 5: The accuracy evaluation results

the NF, FN, and N2F1 models for both the training and testing data. This is likely because the structure of the F2N1 and N3F2 models contains the same Wiener-Hammerstein structure as the target system, while the other models do not. Note that the structure of the target system is unknown in real applications. Therefore, using a deep structure is valuable because it allows the adoption of a model structure that encompasses the true underlying structure of the target system.

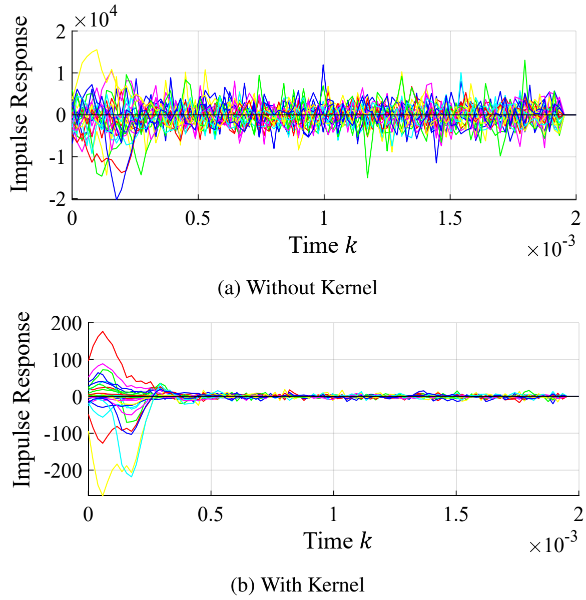
When comparing the results with and without regularization in Fig. 5, it can be observed that incorporating regularization leads to improved accuracy on the testing data.

The finite impulse responses of the models are shown in Figs. 6a and 6b, representing models without and with kernel regularization, respectively.This suggests that the identified model inherits the stability of the target system, which could be considered as a reason for the higher accuracy observed in Fig. 5. The numerical example confirms the effectiveness of the proposed method in improving model accuracy.

6. CONCLUSION

This paper proposed a nonlinear system identification method that incorporates kernel-based regularization into a DNN combining 3-layer NNs and FIR models, and evaluates its effectiveness. In particular, the exponential decay of the finite impulse response in the FIR model components is considered a desirable characteristic from the perspective of design and analysis in the field of control engineering.

In the future, the proposed modeling method will be adopted for the identification of other nonlinear systems, and its accuracy will be verified. This will help elucidate the generalizability and applicability of the proposed model. Furthermore, the proposed model might harbor the potential for a physical interpretation of characteristics in systems where the true structure is unknown. Future work includes clarifying the contributions in this perspective.



(a) Without Kernel

(b) With Kernel

Fig. 6: Finite Impulse Response in The Proposed Model

REFERENCES

[1] L. Ljung, *System Identification: Theory for the User*. Prentice Hall, 1999.

[2] G. Pillonetto and G. De Nicolao, "A new kernel-based approach for linear system identification," *Automatica*, vol. 46, no. 1, pp. 81–93, 2010.

[3] G. Pillonetto, F. Dinuzzo, T. Chen, G. De Nicolao, and L. Ljung, "Kernel methods in system identification, machine learning and function estimation: A survey," *Automatica*, vol. 50, no. 3, pp. 657–682, 2014.

[4] 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.

[5] G. Pillonetto, A. Aravkin, D. Gedon, L. Ljung, A. H. Ribeiro, and T. B. Schön, "Deep networks for system identification: a survey," *arXiv preprint arXiv:2301.12832*, 2023.

[6] F. Giri and E.-W. Bai, *Block-oriented Nonlinear System Identification*. Springer, Aug. 2010.

[7] S. Bai, J. Z. Kolter, and V. Koltun, "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling," *arXiv preprint arXiv:1803.01271*, 2018.

[8] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.

[9] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," *arXiv preprint arXiv:1912.01703*, 2019.

Manuscript 257 submitted to 2024 SICE Festival 2024 with Annual Conference.
Received April 22, 2024.