Development of Wiener-Hammerstein Models Parameterized using Orthonormal Basis Filters and Deep Neural Network

Janak M. Patel * **Kunal Kumar** ** **Sachin C. Patwardhan** *

* *Department of Chemical Engineering, Indian Institute of Technology Bombay, Mumbai, India (e-mail: sachinp@iitb.ac.in)*
** *Department of Systems and Control Engineering, Indian Institute of Technology Bombay, Mumbai, India*

Abstract: Many chemical or biochemical processes exhibit strongly nonlinear dynamic behavior in the desired region of operations. To develop effective monitoring and control schemes for such systems, it is necessary to develop a reliable model that captures dynamics as well as the steady-state behavior over a wide range of operations. In this work, it is proposed to develop a block-oriented Wiener-Hammerstein model parameterized using generalized orthonormal basis filters and deep neural network (GOBF DNN). A two-step procedure is developed to select the generalized orthonormal basis filters (GOBF) pole locations and estimate the deep neural network (DNN) parameters. The efficacy of the proposed modeling strategy is demonstrated using the simulation study on a benchmark continuously operated fermenter system. The proposed GOBF DNN model is able to capture the dynamic and steady-state behavior of the plant over a wide range of operations. Comparison of performances based on the dynamic as well as the steady-state indices clearly underscores the advantages of using a DNN over a shallow neural net and a NARX model developed using DNN.

Keywords: Block Oriented Model, Weiner-Hammerstein Model, Deep Neural Network, Generalized Orthonormal Basis Filters

1. INTRODUCTION

Many chemical or biochemical processes exhibit strongly nonlinear dynamic behavior. In particular, output multiplicities (occurrence of multiple steady states for a fixed set of input values) or input multiplicities (occurrence of identical output steady states for multiple sets of input values) can be observed in reactor systems involving exothermic reactions. Nonlinear control schemes, such as nonlinear model predictive control or nonlinear controller synthesized using input-output linearization, have been suggested in the literature for achieving efficient control of such systems (Henson et al. [1997]). The availability of a dynamic model that can accurately capture the dynamic as well as nonlinear steady-state behavior of the system under consideration is critical for the success of any such nonlinear control scheme.

The block-oriented nonlinear models with Wiener, Hammerstein, or Wiener-Hammerstein structures are frequently used for control relevant black box model development. Because of their relatively simple structure, these models are viewed as 'next-step-beyond-linear modeling' (Pearson and Ogunnaike [1997]). In particular, many variants of block-oriented models have been developed using generalized orthonormal basis filters to parameterize the linear dynamic component (Patwardhan [2014]). McArthur [2012] have proposed the use of Hammerstein and Wiener-Hammerstein models parameterized using GOBF for industrial-scale systems. The nonlinear static maps in the block-oriented models are constructed using ordinal splines. Dasgupta and Patwardhan [2010] have shown that GOBF based Wiener, Hammerstein, or Wiener-Hammerstein models, can be used for capturing the dynamics and nonlinear steady-state behavior of systems exhibiting input multiplicities.

The choice of functions used to parameterize the static blocks in the block-oriented model plays an important role in the dynamic and steady-state behavior of the identified models. Many available approaches use polynomial maps for parameterizing the static blocks. However, this choice may not be adequate for capturing the static nonlinearities in the system over a wide operating range (Dasgupta and Patwardhan [2010]). Alternatively, GOBF-Wiener models have been developed using artificial neural networks for parameterizing the static nonlinearities. Sentoni et al. [1998] initially parameterize the linear dynamics using GOBF while the state to output map is constructed using a single hidden layer neural network. While SNN appears to be a better choice for the static blocks than the polynomial maps, their utility needs to be examined by considering dynamic as well as the steady-state behavior of the resulting model.

In recent years, there has been increasing use of Deep Neural Networks (DNN) for developing control relevant nonlinear dynamic models as DNN has been shown to be a better choice than SNN for capturing nonlinear behavior. Mckay et al. [2021] have used DNN for capturing dynamics of a distributed parameter electrochemical system (Lithium-ion battery). In this work, we propose to use of DNN for the development of MIMO Wiener-Hammerstein (WH) models parameterized using GOBF. The proposed GOBF-WH-DNN model is developed in two steps. To begin with, a GOBF-WH models are developed using a quadratic polynomial map. The pole locations of the GOBF are fixed using this step, and the state sequence generated using the linear dynamic component of the GOBF-WH model serves as an input to the DNN model. Thus, the nonlinear state to output map in the GOBF-WH model is replaced by a DNN to arrive at GOBF-WH-DNN model. The efficacy of the proposed approach is evaluated by conducting simulation studies on a benchmark continuous fermenter system (Henson et al. [1997], Dasgupta and Patwardhan [2010]). Dynamic as well as the steady-state characteristics of the resulting model is used for the model validation exercise. The ability of the proposed approach to capture system behavior is compared with a GOBF-WH model developed using SNN and a NARX model developed using DNN.

This paper is organized into four sections. The development of GOBF-WH-DNN model is presented in Section 2. In section 3, the performance of the proposed model is demonstrated using a simulation study on a continuous fermenter system. The major conclusion drawn from the analysis of the simulation study is presented in section 4.

2. DEVELOPMENT OF BLACK BOX NONLINEAR DYNAMIC MODEL

Consider a continuously operated non-linear process, governed by a mechanistic model of the form:

$$\frac{d\mathbf{Z}}{dt} = F(\mathbf{Z}(t), \mathbf{U}(t), \mathbf{D}(t)) \tag{1}$$

where $\mathbf{Z}(t) \in R^s$ represents state vector, $\mathbf{U}(t) \in R^m$ represents manipulative input vector and $\mathbf{D}(t) \in R^d$ represents unmeasured disturbances vector. The discrete time measurements available from the system (with a regular sampling interval of T) are represented as follows

$$\mathbf{Y}(k) = G(\mathbf{Z}(k)) + \mathbf{v}(k) \tag{2}$$

where $\mathbf{Y}(k) \in R^r$ represents measured output vector corrupted with measurement noise $\mathbf{v}(k)$, which is zero mean random variable. In manipulated inputs are assumed to be piecewise constant injected at a regular sampling interval (T). For the identification of a black box dynamic model, the plant is perturbed deliberately by introducing multi-level perturbations in the manipulated inputs and the corresponding input-output data are collected. The inputs are manipulated at discrete time instants$\{t_k = kT : k = 0, 1, 2, ...\}$ (T is sampling time) and it is assumed that the measurements of the outputs are available at same sampling instants. Therefore, sequence of input data sets $\mathcal{U}_N \equiv \{\mathbf{U}(k) : k = 0, 1, 2, ..., N\}$ and the corresponding output data sets $\mathcal{Y}_N \equiv \{\mathbf{Y}(k) : k = 0, 1, 2, ..., N\}$ are collected from the plant. The modeling exercise is aimed at developing a MIMO discrete time black-box model of the form:

$$\mathbf{Y}(k) = [\mathbf{\Omega}(k), \boldsymbol{\theta}] + \mathbf{e}(k) \tag{3}$$

using sets $\mathcal{Y}_N$ and $\mathcal{U}_N$. Here, $\{\mathbf{\Omega}(k) : k = 1, 2, ...N\}$ represent the regressor vectors and $\{\mathbf{e}(k) : k = 1, 2, ...N\}$ represent sequence of model residuals . If regressor vector $\mathbf{\Omega}(k)$ is chosen as function of past known inputs and past output measurements, then structure of model is referred to as nonlinear ARX (i.e. NARX) and if it is chosen as function of past known inputs and simulated outputs from past input only, then structure of model is referred to as Nonlinear output error (or NOE) (Dasgupta and Patwardhan [2010]). This work aims at development of a MIMO model with NOE structure.

2.1 Deep Neural Network

In this subsection, a deep neural net (DNN) is introduced as algebraic (static) maps connecting an input $\mathbf{X} \in R^n$ with an output $\mathbf{Y} \in R^s$. The connection of DNN with the proposed dynamic models will be discussed later. Thus, consider deep neural network of p hidden layer described as follows (Emmert-Streib et al. [2020]). Let $\mathbf{W}^{(i)}$ represent the weight matrix associated with the i^{th} hidden layer and let $\mathbf{b}^{(i)}$ represents the bias vector associated with the i^{th} hidden layer where $i = 1, 2, ..p$. Then, defining $\boldsymbol{\eta}^{(0)} = \mathbf{X}$, the intermediate outputs of hidden layers can be represented as follows:

$$\boldsymbol{\eta}^{(i)} = \Psi^{(i)}(\mathbf{W}^{(i)}\boldsymbol{\eta}^{(i-1)} + \mathbf{b}^{(i)}) \text{ for } i = 1, 2, ..p \tag{4}$$

where $\Psi^{(i)}$ represents the activation function of i^{th} hidden layer. There are various types of activation functions suggested in the literature, such as *sigmoidal,* tanh*, linear etc.,* that can be used to construct transformation $\Psi^{(i)}$(Emmert-Streib et al. [2020]). The output Y is related to $\boldsymbol{\eta}^{(p)}$ by a linear relation as follows

$$\text{Output layer: } Y = \mathbf{W}^{(p+1)}\boldsymbol{\eta}^{(p)} + \mathbf{b}^{(p+1)} + \mathbf{e} \tag{5}$$

where $\mathbf{e}$ represents the approximation error vector. Thus, the output is related to the input as follows:

$$Y = NN\,(X, \boldsymbol{\theta}_{NN}) + \mathbf{e} \tag{6}$$

where

$$NN\,(X) \equiv \mathbf{W}^{(p+1)}\Psi^{(p)}\{\Psi^{(p-1)} \\ (...\Psi^{(1)}(\mathbf{W}^{(1)}X + \mathbf{b}^{(1)}))\} + \mathbf{b}^{(p+1)} \tag{7}$$

Here, $\boldsymbol{\theta}_{NN}$ refers to NN parameters $\{\mathbf{W}^{(i)}, \mathbf{b}^{(i)} : i = 1, 2, , , p + 1\}$. In this work, if p is chosen equal to 1, then the resulting neural net is referred to as a shallow neural network (SNN) and if number of hidden layer is more than one, then it's referred to as a deep neural network (DNN). Now, for a given set of input vectors, $\mathcal{X}_N$, and associated output vectors, $\mathcal{Y}_N$

$$\mathcal{X}_N = \left\{ \mathbf{X}^{(0)},\ \mathbf{X}^{(1)},\ ...\ \mathbf{X}^{(N)} \right\}$$

$$\mathcal{Y}_N = \left\{ \mathbf{Y}^{(0)},\ \mathbf{Y}^{(1)},\ ...\ \mathbf{Y}^{(N)} \right\}$$

the identification problem is to estimate the unknown parameter weights and bias (i.e. $\boldsymbol{\theta}_{NN}$) such that some scalar function of

$$\mathbf{e}^{(k)} = \mathbf{Y}^{(k)} - NN\left(\mathbf{X}^{(k)}, \boldsymbol{\theta}_{NN}\right) \text{ for } k = 0, 1, 2, ..., N \tag{8}$$

is minimized.

2.2 Linear Dynamic Component Modeling using GOBF

Let $u_j(k) = U_j(k) - \overline{U_j}$ for $j = 1, 2, ..., m$ represent perturbation inputs and let $y_j(k) = Y_j(k) - \overline{Y}_j$ for $j = 1, 2, ..., r$ represent perturbation outputs where $\overline{U}$ and $\overline{Y}$ represent some nominal steady state input and output, respectively. Consider a MISO model relating inputs with i'th output (To simplify the notations, we drop subscript i from $y_i(k)$).

$$y(k) = \sum_{j=1}^{m} G_j(z) u_j(k) \tag{9}$$

where $G_j(z)$ represents the strictly stable transfer function. Then, the each $G_j(z)$ can be expressed as

$$G_j(z) \approx \sum_{l=1}^{n} c_{jl} F_{jl}(z, \boldsymbol{\xi}^{(j)}) \tag{10}$$

where $F_{jl}(z, \boldsymbol{\xi}^{(j)})$ represents an orthonormal basis for the set of strictly proper stable transfer functions (denoted as $\mathcal{H}_2$). A complete orthogonal set in $\mathcal{H}_2$ can be constructed as follows Patwardhan and Shah [2004]:

$$F_{j,l}(z, \boldsymbol{\xi}^{(j)}) = \frac{\sqrt{(1 - \left|\xi_l^{(j)}\right|^2)}}{\left(z - \xi_l^{(j)}\right)} \prod_{s=1}^{l-1} \frac{\left(1 - \xi_s^{(j)*} z\right)}{\left(z - \xi_s^{(j)}\right)} \tag{11}$$

where $\boldsymbol{\xi}^{(j)}$ is a vector of poles inside the unit circle appearing in complex conjugate pairs. Patwardhan and Shah [2004] have shown a state-space realization of linear MISO model parametrized using GOBF can be constructed as follows:

$$\mathbf{x}(k+1) = \Phi \mathbf{x}(k) + \Gamma \mathbf{u}(k) \tag{12}$$

$$y(k) = C\mathbf{x}(k) + e(k) \tag{13}$$

where the matrices Φ and Γ are only function of GOBF poles (i.e. $\boldsymbol{\xi}^{(j)} : j = 1, 2, .., m$) and vector C consists of only the Fourier coefficients c_{jl} (Patwardhan and Shah [2004]). For MIMO system with r outputs, r such MISO models can be stacked to arrive at a MIMO state space model of the form (Patwardhan and Shah [2004])

$$\mathbf{X}(k+1) = \boldsymbol{\Phi} \mathbf{X}(k) + \boldsymbol{\Gamma} \mathbf{u}(k) \tag{14}$$

$$\mathbf{y}(k) = \mathbf{C}\mathbf{X}(k) + \mathbf{e}(k) \tag{15}$$

where

$$\mathbf{y}(k) = \left[\, y_1(k) \ y_2(k) \ ... \ y_r(k) \,\right]^T$$
$$\mathbf{X}(k) = \left[\, \mathbf{x}^{(1)}(k)^T \ \mathbf{x}^{(2)}(k)^T \ ... \ \mathbf{x}^{(r)}(k)^T \,\right]^T$$
$$\boldsymbol{\Phi} = blockdiag \left[\, \Phi^{(1)} \ \Phi^{(2)} \ ... \ \Phi^{(r)} \,\right]$$
$$\boldsymbol{\Gamma} = \left[\, \Gamma^{(1)} \ \Gamma^{(2)} \ ... \ \Gamma^{(r)} \,\right]^T$$
$$\mathbf{C} = blockdiag \left[\, C^{(1)} \ C^{(2)} \ ... \ C^{(r)} \,\right]$$

2.3 Development of Wiener-Hammerstein-DNN Model

It is proposed to develop Wiener-Hammerstein model parameterized using generalized orthonormal basis filters (GOBF) (Dasgupta and Patwardhan [2010]) and deep neural networks which has following generalized form:

$$\mathbf{X}(k+1) = \boldsymbol{\Phi} \mathbf{X}(k) + \boldsymbol{\Gamma} \Lambda[\mathbf{u}(k)] \tag{16}$$

$$\mathbf{y}(k) = \Omega\left[\, \mathbf{X}(k) \right] + \mathbf{e}(k) \tag{17}$$

It should be noted that $\Lambda[.]$ represents nonlinear map between manipulative inputs and states as follows:

$$\boldsymbol{\Lambda}(\mathbf{u}(k)) = \left[\, f_1\left[\mathbf{u}(k)\right] \ f_2\left[\mathbf{u}(k)\right] \ ... \ f_M\left[\mathbf{u}(k)\right] \,\right]^T \tag{18}$$

Here, $f_j\left[\mathbf{u}(k)\right] : R^m \rightarrow R$ for $j = 1, 2, ..., M$ represent some nonlinear functions of inputs. In this work, these are chosen as polynomial functions as follows:

$$f_i\left[\mathbf{u}(k)\right] = u_i(k) \text{ for } i = 1, 2, ...m \tag{19}$$

$$f_{(m+l)}\left[\mathbf{u}(k)\right] = u_i(k) u_j(k) \tag{20}$$

for $i, j = 1, 2, ...m$ and $l = 1, 2, ..., m(m+1)/2$

The nonlinear map between states and measured outputs (i.e. $\Omega[.]$) is taken as deep neural network as following

$$\Omega\left[\, \mathbf{X}(k) \right] = NN\left[\, \mathbf{X}(k), \boldsymbol{\theta}_{NN} \right] \tag{21}$$

It is to be noted that the resulting model has NOE structure and has better long range prediction abilities. Since all eigenvalues of Φ are strictly inside the unit circle by construction, it is straight forward to investigate the steady state behavior of the resulting WH-GOBF-DNN model. For any given steady state input $\mathbf{u}_s$, steady state output can be calculated analytically as following

$$\mathbf{X}_s = [\mathbf{I} - \boldsymbol{\Phi}]^{-1} \boldsymbol{\Gamma} \Lambda[\mathbf{u}_s] \tag{22}$$

$$\mathbf{y}_s = NN[\mathbf{X}_s] \tag{23}$$

The unknown parameters of the proposed model are $\boldsymbol{\theta}_{OBF} \equiv \{\boldsymbol{\xi}^{(i,j)} : i = 1, 2, ...r \text{ and } j = 1, 2, ..., m\}$ and weights $\boldsymbol{\theta}_{NN}$. Given data sets $\mathcal{U}_N \equiv \{\mathbf{u}(k) : k = 0, 1, 2, ..., N\}$ and $\mathcal{Y}_N \equiv \{\mathbf{y}(k) : k = 0, 1, 2, ..., N\}$, In this work we propose to identify these parameters sequentially.

- In the first step, we identify MISO GOBF-Wiener-Hammerstein models of the form

$$\mathbf{x}^{(i)}(k+1) = \boldsymbol{\Phi}^{(i)} \mathbf{x}^{(i)}(k) + \boldsymbol{\Gamma} \Lambda\left[\mathbf{u}(k)\right] \tag{24}$$

$$y_i(k) = \mathcal{Q}^{(i)}\left[\mathbf{x}^{(i)}(k)\right] + e_i(k) \tag{25}$$

  where $\mathcal{Q}^{(i)}[.]$ represents quadratic polynomial output maps as described by Dasgupta and Patwardhan [2010], using $\{\mathcal{U}_N, \mathcal{Y}_N\}$. This step yields estimates of $\boldsymbol{\theta}_{OBF}$, say $\widehat{\boldsymbol{\theta}}_{OBF}$.
- In the next step, we construct the set $\mathcal{X}_N \equiv \{\mathbf{X}(k) : k = 0, 1, 2, ...N\}$ as follows

$$\mathbf{X}(k+1) = \boldsymbol{\Phi}\left(\widehat{\boldsymbol{\theta}}_{OBF}\right) \mathbf{X}(k) + \boldsymbol{\Gamma}\left(\widehat{\boldsymbol{\theta}}_{OBF}\right) \Lambda[\mathbf{u}(k)] \tag{26}$$

  where matrices (Φ, Γ) are constructed using $\widehat{\boldsymbol{\theta}}_{OBF}$. Then, a deep neural net is developed using $\mathcal{X}_N$ as the input set and $\mathcal{Y}_N$ as the output, i.e.

$$\mathbf{y}(k) = NN[\,\mathbf{X}(k), \boldsymbol{\theta}_{NN}] + \mathbf{e}(k) \tag{27}$$

  This step yields estimates of $\boldsymbol{\theta}_{NN}$. The deep neural network is developed by selecting appropriate number of hidden layers, number of neurons in each hidden layer, and activation function for each layers.

2.4 Development of NARX Model using Neural Networks

Another popular form of neural network based nonlinear discrete time model is with nonlinear ARX (NARX) structure (Bhat and McAvoy [1989]). Thus, for the purpose of comparison, it is suggested to develop a MIMO NARX model using deep neural networks, which has the following generalized form:

$$\mathbf{Y}(k) = NN[\mathbf{Y}(k-1), ..., \mathbf{Y}(k-d), \mathbf{U}(k-1), \\ ..., \mathbf{U}(k-d), \boldsymbol{\theta}_{NN}] + \mathbf{e}(k) \quad (28)$$

where d represents number of delayed measurements and inputs used for developing the model. Given data sets $\{\mathcal{U}_N, \mathcal{Y}_N\}$, parameters of the DNN based NARX model are identified using the back-propagation algorithm. For the purpose of comparison with the proposed NOE model, the following behavior of the NARX model will be investigated

- **Model simulation:** For a given sequence of inputs, $U(0), U(1), ...$ the simulated model output is computed as
$$\widehat{\mathbf{Y}}(k) = NN[\widehat{\mathbf{Y}}(k-1), .., \widehat{\mathbf{Y}}(k-d), \\ \mathbf{U}(k-1), .., \mathbf{U}(k-d)] \quad (29)$$
where $\widehat{\mathbf{Y}}$ represents the predicted output.
- **Steady State behavior:** For the NARX Model, it's not possible to estimate the steady states analytically. For a specified steady state input, say $\mathbf{U}_s$, the corresponding steady state $\mathbf{Y}_s$ is computed by simulating output as
$$\widehat{\mathbf{Y}}(k) = NN[\widehat{\mathbf{Y}}(k-1), .., \widehat{\mathbf{Y}}(k-d), \mathbf{U}_s, .., \mathbf{U}_s] \quad (30)$$
till $\widehat{\mathbf{Y}}(k) \rightarrow \mathbf{Y}_s$.

3. SIMULATION STUDY

Simulation study on a benchmark continuous fermenter system (Henson et al. [1997], Dasgupta and Patwardhan [2010]) is conducted in this section. This system exhibits input multiplicities and change in the sign of the steady-state gain(s) in the desired operating region. The identification of three NN based models is carried out using an input-output data set generated by simulating the dynamic system behavior over a wide range of operations. Two Wiener Hammerstein models parameterized using GOBF are developed using this data: (i) WH model with a deep NN as the state to output map (referred to as GOBF-DNN) and (ii) WH model with a shallow NN as the state to output map (referred to as GOBF-SNN). In addition, a NARX model parameterized using DNN (referred to as a NARX-DNN) is developed using the data set. The performances of Wiener-Hammerstein and NARX models are compared using the following performance indices

- **Dynamic Sum Squared Error (DSSE):**
$$DSSE = \sum_{k=1}^{N} \left[\mathbf{Y}(k) - \widehat{\mathbf{Y}}(k)\right]^T \left[\mathbf{Y}(k) - \widehat{\mathbf{Y}}(k)\right] \quad (31)$$
Here, N represents the number of data points in the model validation data set, $\mathbf{Y}(k)$ represents the process output vector and $\widehat{\mathbf{Y}}(k)$ represents the simulated model output vector.
- **Static Sum Squared Error (SSSE):**
$$SSSE = \sum_{i=1}^{Ns} \left[\mathbf{Y}_s - \widehat{\mathbf{Y}}_s\right]^T \left[\mathbf{Y}_s - \widehat{\mathbf{Y}}_s\right] \quad (32)$$
Here, $\mathbf{Y}_s$ represents the steady state process output vector and $\widehat{\mathbf{Y}}_s$ represents the model steady state output vector obtained for a fixed $\mathbf{U}_s$ and $\mathbf{N}_s$ represents the number of steady state data points.

The simulation study is carried out in MATLAB 2021 on a standard PC with i7 processor. Also, MATLAB *nnstart* toolbox has been used to train neural network models.

3.1 System Description

The nonlinear dynamics of the fermentation process (Henson et al. [1997], Dasgupta and Patwardhan [2010]) is given by the following set of ODE equations:

$$\frac{dX}{dt} = -DX + \mu(P, S)X \quad (33a)$$
$$\frac{dS}{dt} = D(S_f - S) - \frac{1}{Y_{x/s}}\mu(P, S)X \quad (33b)$$
$$\frac{dP}{dt} = -DP + (\alpha\mu(P, S) + \beta)X \quad (33c)$$

where X, S, P represents the biomass concentration, substrate concentration, and product concentration respectively, $\mu(P, S)$ represents specific growth rate, $Y_{x/s}$ represents cell mass yield and (α, β) are assumed to be constant i.e. independent of the operating conditions. The dilution rate (D denoted as U_1) and feed substrate concentration (S_f denoted as U_2) are manipulated inputs while biomass concentration (X) and product concentration (P) are measured outputs. The specific growth rate ($\mu(P, S)$), exhibits both substrate and product inhibition, has the following form

$$\mu(P, S) = \frac{\mu_m \left(1 - \frac{P}{P_m}\right) S}{K_m + S + \frac{S^2}{K_i}} \quad (34)$$

Here, μ_m, P_m, K_m represents the maximum growth rate, product saturation constant, and inhibition constant respectively. The details of the model parameters can be found in Henson et al. [1997]. The measurements are collected at a sampling interval (T) of 0.1 hr. The nominal steady state operating condition for computation of perturbation variables is chosen as the peak point that corresponds to $D = 0.16$ *and* $S_f = 23.4$,i.e.

$$\mathbf{Y}_s = [\, 7.3059 \;\; 25.008 \,]^T$$

The output measurements are corrupted with zero mean Gaussian white noise with standard deviations 0.1 and 0.3. The process is perturbed by introducing multi-level PRBS signals (ref. Fig.1) in both the manipulated inputs simultaneously and 9900 data points (equivalent to 990 hours of plant run) are collected. The collected output data set used for the system identification exercise. It can be observed that the data covers a wide range of operation and can be viewed as collection of perturbation studies conducted in the neighborhood of multiple steady operating conditions.

3.2 Black Box Nonlinear Dynamic Models

For the development of the proposed GOBF-DNN and GOBF-SNN models, two MISO Wiener-Hammerstein models of the form (24)-(25) are developed. The input nonlinear map $\Lambda[.]$ in the Wiener-Hammerstein models are taken as 5×1 vector

$$\Lambda[\mathbf{u}(k)] = [u_1(k),\ u_2(k),\ u_1^2(k),\ u_2^2(k), \\ u_1(k)u_2(k)] \quad (35)$$

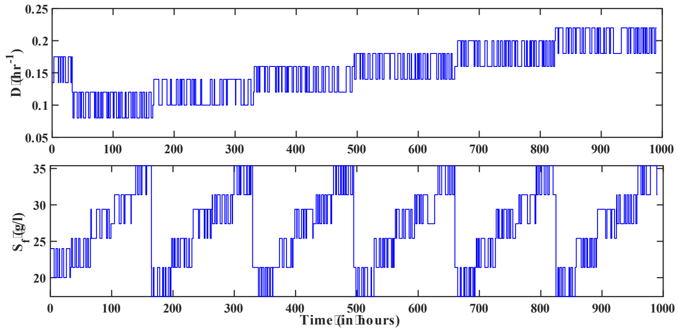

Fig. 1. Multi-level PRBS input sequence

Table 1. Optimum GOBF poles of Wiener-Hammerstein Model

Input	MISO-1 (X)	MISO-2 (P)
u_1	[0.1825 0.0557]	[0.1043 0.0514]
u_2	[0.0804 0.0804]	[0.0726 0.0726]
u_1^2	[0.0256]	[0.0245]
u_2^2	[0.0737]	[0.0513]
$u_1 u_2$	[0.0627]	[0.0543]

Table 2. Hyper-parameter used for the development of Neural Networks

Model ↓	p	N	Ψ
NARX-DNN	3	6,12,4	"tanh"
GOBF-SNN	1	10	"tanh"
GOBF-DNN	4	4,8,4,2	"tanh"

Initially two MISO Wiener-Hammerstein models with 7 GOBF poles each are identified using the input-output data set. The optimum location of equivalent continuous time poles, ζ_i ,identified from data are mentioned in the Table 1. These poles are related to the GOBF poles as $\xi_i = \exp(-\zeta_i T)$. These pole locations are then used to construct Φ, Γ matrices in eq. (26) to generate the state sequence for training of DNN or SNN. The development of neural networks requires the tuning of a number of hyperparameters such as number of hidden layer (p), number of neurons (N) in each hidden layer and activation function (Ψ) for each hidden layer that are listed in Table 2. Since, the continuous fermenter is a 3^{rd} order system, it is proposed to develop MIMO NARX model using a deep neural net subjected to inputs with 3 delays, i.e. d = 3 in eq. (28). The number of unknown parameters which are being estimated for each model are as follow

$$\text{GOBF-DNN: } 14 \text{ (GOBF)} + 152 \text{ (DNN)} = 166$$

$$\text{GOBF-SNN: } 14 \text{ (GOBF)} + 172 \text{ (DNN)} = 186$$

$$\text{NARX-DNN: } 224 \text{ (DNN)}$$

3.3 Results and Discussion

The simulation (or infinite horizon prediction) performance of the identified models is validated using an independent dynamic data set as shown in Figure 2, while the steady-state behavior (in the neighborhood of the nominal operating condition) is compared in Figures 3 and 4. Performance comparison in terms of the performance indices is presented in Table 3. From the Figure 2, it is observed that the performance of the GOBF-DNN model is the best amongst the three while the performance of the GOBF-SNN model is the poorest. This is also reflected in the DSSE values reported in Table 3. Although the NARX-DNN model is able to capture the dynamic behavior of the plant reasonably well, it fails to capture the steady-state behavior of the system. On the other hand, the GOBF-DNN model is able to capture the steady-state behavior of the plant over a wide range of operations. The GOBF-SNN model captures the steady-state behavior of the plant at a high feed substrate concentration or high dilution rate. But performs poorly for low S_f values. For further investigation of steady-state behavior of these GOBF-DNN and GOBF-SNN models, a 3D plot of steady-state outputs as a function of substrate concentration are plotted for different dilution rates as shown in Figures 5 and 6. The SSSE values corresponding to Fig. 5 and Fig. 6 are given in Table 4. Except for the high dilution rate, Figures 5 and 6 suggests that the GOBF-DNN model is able to capture the steady-state behavior of the plant over a wide range of operation.

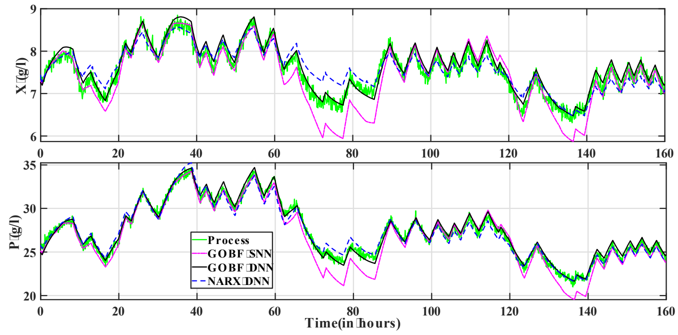

Fig. 2. Dynamic model validation

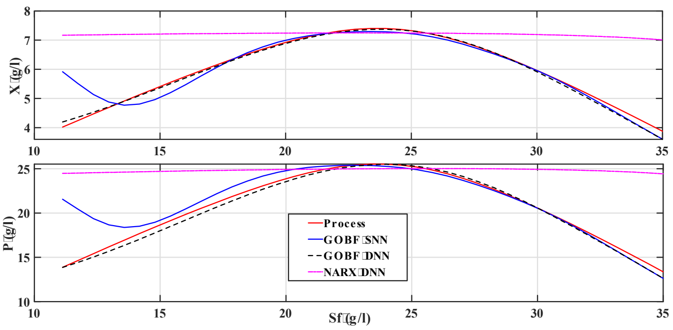

Fig. 3. Comparison of steady state behavior w.r.t. S_f at $D = 0.16$

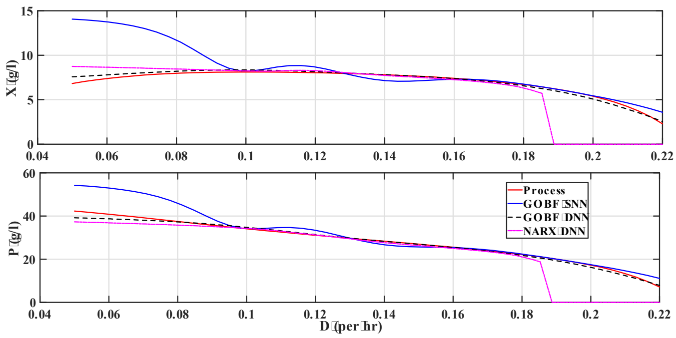

Fig. 4. Comparison of steady state behavior w.r.t. D at $S_f = 23.4$

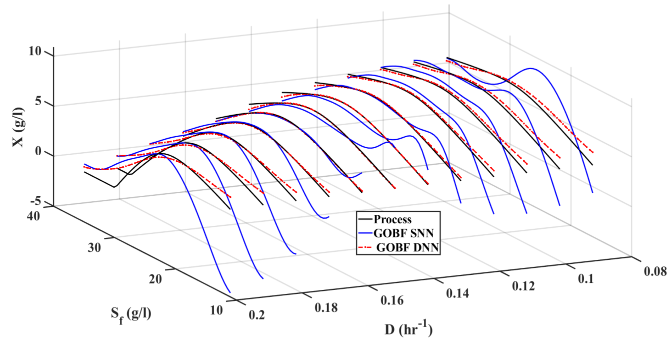

Fig. 5. Comparison of steady state behavior of X

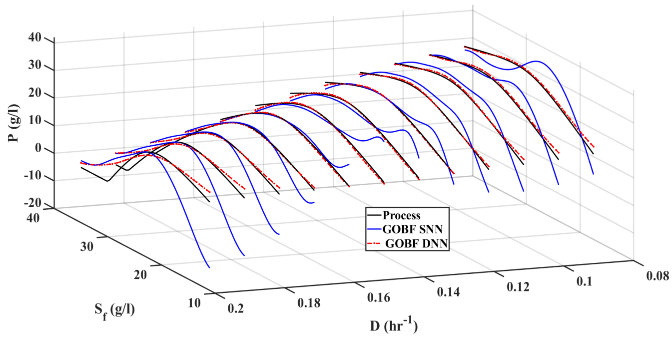

Fig. 6. Comparison of steady state behavior of P

Table 3. Comprison of dynamic and steady-state behaviour of Models using SSE

Model ↓	DSSE	SSSE (w.r.t. S_f)	SSSE (w.r.t. D)
NARX DNN	583.5634	1337.9	2785.1
GOBF SNN	1764.3	134.4925	1735.2
GOBF DNN	351.9407	6.3845	48.0078

Table 4. Comparison of steady-state behaviour w.r.t. dilution rate using SSSE

$D(h^{-1})$	GOBF SNN	GOBF DNN
0.09	1574.1	59.5476
0.10	713.6664	49.2099
0.11	646.0348	28.2202
0.12	544.3256	10.6538
0.13	371.5349	6.4298
0.14	1159.2	13.5248
0.15	1421.5	17.8227
0.16	178.6279	7.9081
0.17	561.0108	10.6411
0.18	2500.9	134.7279
0.19	4456.5	369.8197
0.2	5496.7	355.3697

4. CONCLUSION

In this work, we developed block-oriented Wiener Hammerstein models parameterized using generalized orthonormal basis filters (GOBF) using input-output data. The nonlinear state to output map in a Wiener-Hammerstein model is constructed using a deep neural network. The resulting GOBF-DNN model has a nonlinear output error structure, and, as a consequence, is better suited for carrying out long-range predictions. Since the dynamic component is constructed using discrete-time transfer functions, the resulting model can be viewed as a grey-box model when compared with a NARX type black-box model developed using DNN. The efficacy of the proposed model is demonstrated using a simulation study on a benchmark continuous fermenter process, which exhibits input multiplicity behavior. The simulation study compares the model simulation performances of the proposed GOBF-DNN with GOBF-SNN and NARX-DNN models. The analysis of the simulation study reveals that the NARX-DNN model shows good dynamic simulation performance. However, it fails to capture the steady-state characteristics of the system. On the other hand, the proposed GOBF-DNN model is able to capture the dynamic and steady-state behavior of the plant over a wide range of operations. Comparison of performances based on dynamic as well as the steady-state indices clearly underscores the advantages of using a DNN over an SNN for constructing the state to output map.

REFERENCES

Bhat, N., & McAvoy, T. J. (1990). Use of neural nets for dynamic modeling and control of chemical process systems. Computers & Chemical Engineering, 14(4-5), 573-582.

Dasgupta, D., & Patwardhan, S. C. (2010). NMPC of a continuous fermenter using Wiener-Hammerstein model developed from irregularly sampled multi-rate data. IFAC Proceedings Volumes, 43(5), 637-642.

Emmert-Streib, F., Yang, Z., Feng, H., Tripathi, S., & Dehmer, M. (2020). An introductory review of deep learning for prediction models with big data. Frontiers in Artificial Intelligence, 3, 4.

Henson, M. A., & Seborg, D. E. (1997). Nonlinear process control (pp. 5-8). Upper Saddle River, New Jersey: Prentice Hall PTR.

Kumar, K. K., & Patwardhan, S. C. (2002). Nonlinear predictive control of systems exhibiting input multiplicities using the multimodel approach. Industrial & engineering chemistry research, 41(13), 3186-3198.

MacArthur, J. W. (2012). A new approach for nonlinear process identification using orthonormal bases and ordinal splines. Journal of Process Control, 22(2), 375-389.

Mckay, M. B., Wetton, B., & Gopaluni, R. B. (2021). Learning physics based models of Lithium-ion Batteries. IFAC-PapersOnLine, 54(3), 97-102.

Patwardhan, S. C.(2014). Development of Nonlinear Black Box Models using Orthonormal Basis Filters: A Review. Proc. of International Symposium on Advanced Control of Industrial Processes 2014, Hiroshima, Japan, 28-30 May, 2014.

Patwardhan, S. C., & Shah, S. L. (2005). From data to diagnosis and control using generalized orthonormal basis filters. Part I: Development of state observers. Journal of Process control, 15(7), 819-835.

Pearson, R. K., & Ogunnaike, B. A. (1997). Nonlinear process identification. Nonlinear process control, 11-109.

Sentoni, G. B., Biegler, L. T., Guiver, J. B., & Zhao, H. (1998). State-space nonlinear process modeling: Identification and universality. AIChE Journal, 44(10), 2229-2239.