Computers and Chemical Engineering 159 (2022) 107674

Contents lists available at ScienceDirect

Computers and Chemical Engineering

journal homepage: www.elsevier.com/locate/compchemeng

Technical Note

Compact neural network modeling of nonlinear dynamical systems via the standard nonlinear operator form

Pil Rip Jeon[a,b,c], Moo Sun Hong[c], Richard D. Braatz[c,*]

[a] Department of Chemical and Biomolecular Engineering, Yonsei University, 50 Yonsei-ro, Seodaemun-gu, Seoul, 03722, Republic of Korea
[b] Department of Chemical Engineering, Kongju National University, 1223-24, Cheonan-daero, Seobuk-gu, Cheonan-si, 31080, Republic of Korea
[c] Department of Chemical Engineering, Massachusetts Institute of Technology, Cambridge, 02139, MA, USA

ARTICLE INFO

Article history:
Received 16 May 2021
Revised 6 November 2021
Accepted 16 January 2022
Available online 22 January 2022

Keywords:
Data-driven modeling
Nonlinear model identification
Neural networks
Dynamic artificial neural networks
Bioreactors
Biomanufacturing

ABSTRACT

Dynamic artificial neural networks (DANNs) have become popular for the data-driven modelling of nonlinear dynamical systems. This article elucidates properties of a compact DANN model structure called the standard normal operator form (SNOF). Sets of nonlinear dynamical systems are characterized for which the SNOF can achieve the same model identification error with fewer neurons than needed by the popular DANN model structures. The results are demonstrated in a case study for a multi-stage bioreactor, which is a highly nonlinear dynamical system in which the cells have a sharp change in dynamics during a change in the feed composition as the bioreactor shifts from growth mode to production mode. The ability of the SNOF to model highly nonlinear dynamical systems with a very small number of neurons suggests its potential for serving as a basis for the design of model-based optimal control systems with theoretical guarantees of closed-loop stability and performance.

1. Introduction

The ability of any static nonlinear function to be approximated within any degree of accuracy by an artificial neural network (ANN) initiated their wide application for black-box identification of nonlinear dynamical systems. In particular, dynamical ANNs (DANNs) are the models that have been used in nonlinear control technologies by Rockwell/Pavilion and AspenTech for more than twenty years (Qin and Badgwell, 2003). While DANN-based control algorithms have been developed and applied for decades, the control design and the closed-loop dynamics achieved in industrial applications are not grounded in control theory. More specifically, the industrial control design methods provide no stability or performance guarantees, and the DANN-based control designs can destabilize the closed-loop system. This situation is addressed in industrial applications by carrying out extensive closed-loop simulations over wide variations in operating conditions, and re-tuning the control weights until closed-loop stability is achieved while minimizing oscillations and overshoot. Because any finite number of closed-loop simulations cannot cover all of the potential operating conditions of the system, DANN-based control systems need to be conservatively tuned (that is, to have sluggish performance) to provide strong confidence of their reliability when applied to the real system. This situation is an especial concern for applications in the manufacturing of small-molecule pharmaceuticals and biotherapeutics, in which a deviation from the desired product quality attributes can result in adverse side affects (FDA, 1999).

Many researchers in the academic control communities in the last several decades have worked to provide a theoretical foundation for the stability analysis of DANNs (e.g. see the reviews by Suykens et al., 1996 and Kim et al., 2018 and citations therein). In particular, many sufficient conditions have been derived for the global asymptotic stability of a DANN. In short, the analysis tools tend to be least conservative when the number of neurons is small. More precisely, the number of terms that introduce conservatism in such Lyapunov analysis based on any of the model structures is proportional to the number of neurons. Since a closed-loop system in which the process and controller are each written as DANNs can be written as one large DANN by applying block diagram algebra (e.g., Desoer and Vidyasagar, 1975), these stability conditions are theoretically applicable to both open- and closed-loop systems. The theoretical results that produce the tightest (aka least conservative) conditions apply the Lyapunov method (La Salle and Lefschetz, 1961) to a Lur'e-Postnikov function (Forti and Tesi, 1995) to derive a set of inequalities which, if feasible, implies that the DANN is globally asymptotically stable. These Lyapunov functions consist of a quadratic function plus the sum over the number of neurons of some function of the neuron. Each term in the sum

* Corresponding author at: 77 Massachusetts Avenue, Room E19-551, Cambridge, MA 02139.
E-mail address: braatz@mit.edu (R.D. Braatz).

https://doi.org/10.1016/j.compchemeng.2022.107674
0098-1354/

is subsequently bounded while introducing some conservatism. The introduction of each additional neuron to a DANN results in introducing another conservative step in the Lyapunov analysis, so that the overall stability condition tends to be too conservative to be practically useful for DANNs with a large number of neurons.

For the theoretical results to be sufficiently nonconservative to be practically useful in applications, an important consideration is the number of neurons needed by a DANN model structure to be able to describe the input-output dynamics of the system. The article provides insights into the number of required neurons for all three popular classes of nonlinear dynamic ANN (DANN) model structures: the Neural State-Space Model (NSSM), Global Input-Output Model (GIOM), and Dynamic Recurrent Neural Network (DRNN). The NSSM is a nonlinear dynamical state-space model in which nonlinearities are parameterized by a feedforward ANN (FANN) (Suykens et al., 1995). The GIOM has a recursive input-output structure, in which prediction is made from a finite measurement of past inputs and outputs, and the nonlinearities of the system are parameterized by FANNs (Billings et al., 1992; Levin and Narendra, 1995; Narendra and Parthasarathy, 1990). The structure of the DRNN is nearly the same as the NSSM, but with a linear recursive term added to the state equation for the DRNN (Jin and Gupta, 1996; Fang and Kincaid, 1996). The property that ANNs can approximate any nonlinear function with arbitrarily small error (Cybenko, 1989) can be used to show that all three classes of DANNs can approximate nonlinear dynamical systems within any arbitrarily small error (Kim et al., 2018), making them suitable as a basis for the black-box modeling and control of nonlinear dynamical systems.

After the above background analysis, the article considers the standard nonlinear operator form (SNOF), which is a nonlinear model structure that can be shown to approximate nonlinear dynamical systems with arbitrarily small approximation error (Kim et al., 2018). The analysis motivates the use of the SNOF, instead of the three DANNs, as a theoretically rigorous and practically useful basis for the analysis and control of nonlinear dynamical systems.

Although not the main focus of this article, it is worth mentioning that all three of the aforementioned DANNs can be transformed into an equivalent SNOF, so global asymptotic stability of the three DANNs can be assessed by analyzing the stability of their SNOFs. In particular, any less conservative stability conditions derived for a SNOF can be immediately applied to the three DANNs (Kim et al., 2018). For any of these model structures, the conservatism of applying Lyapunov analysis is based on the number of neurons in a model, which motivates the focus here of comparing the model structures in terms of the number of neurons needed to model a nonlinear dynamical system. The preferred model structure is that which is most "compact," which in this context refers to having the smallest number of neurons needed to fit data with small model error.

A relevant prior result is that an example has been published in which a best-fit SNOF had a smaller model error than best-fit DANNs when all were restricted to have the same number of neurons (Kim et al., 2011). This article provides a deeper structural comparison between the different model structures. We show that the 3 DANNs have highly restrictive model structures compared to the SNOF. Then we derive the number of degrees of freedom in the different model structures for a class of simple nonlinear dynamical systems, which provides enough insight to derive the results for arbitrary numbers of states and neurons. Then DANNs and SNOF are compared in terms of their ability to model the input to state relationships for a multistage bioreactor with highly nonlinear dynamics, which show that the SNOF is able to model the nonlinear dynamic relationships with only one or two neurons.

This paper is organized as follows. Section 2 provides mathematical background and defines nomenclature for ANNs, DANNs, and SNOF. Section 3 analyzes the different DANN model structures in terms of their matrix structure and eigenvalues, after being reformulated as SNOFs. Section 4 derives the number of degrees of freedom for the DANNs and SNOF, illustrating the calculations in simple examples. Section 5 provides the bioreactor case study, and Section 6 concludes the paper.

2. Theoretical background

This section provides some background on ANNs, DANNs, SNOF, and some results in controllability and observability used later in the article.

2.1. Architecture of artificial neural networks

An artificial neural network (ANN) can approximate any nonlinear relationship with bounded input and bounded output with arbitrarily small approximation error (Cybenko, 1989). The basic processing unit of an ANN is called a neuron, the collection of the neurons is called a layer, and the interconnection of the layers constitutes an ANN. The architecture of ANN refers to how the layers are interconnected with each other. A neuron can be represented as $y = \gamma(\sum_{i=1}^{m} w_i s_i + \beta)$ where γ is a nonlinear function referred to as an activation function, $w = [w_1, w_2, \cdots, w_m]^\top$ is a weight vector which represents connection between the neuron and previous layer, s_i are the inputs dispatched from the previous layer to the neuron, and β is a bias term. For the activation function γ, a monotonic function bounded in the interval is usually selected such as hyperbolic tangent function $\in [-1, 1]$ or sigmoid function $\in [0, 1]$. The most popular ANN architecture is the Feedforward ANN (FANN) in which input, hidden, and output layers are connected in series. The activation function of the output layer is usually chosen as a linear function (Levin and Narendra, 1995), and the FANN can be represented as $y = \sum_{i=1}^{h} w_i \gamma(\sum_{j=1}^{m} v_{ij} u_j + \beta_i)$, where v_{ij} represents the weight between the jth neuron in the input layer and the ith neuron in the hidden layer, w_i represents the weight between the jth neuron in the hidden layer and the neuron in the output layer.

2.2. Dynamic artificial neural network model structures

A discrete-time state-space system can be represented as

$$\begin{aligned} x_{k+1} &= f(x_k, u_k) + v_k; \qquad x(0) = x_0 \\ y_k &= g(x_k, u_k) + w_k \end{aligned} \tag{1}$$

where $y \in \mathbb{R}^l$, $u \in \mathbb{R}^m$, $x \in \mathbb{R}^n$, $v \in \mathbb{R}^n$, and $w \in \mathbb{R}^l$ represent the output, input, state, process noise, and measurement noise; $f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ and $g : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^l$ are assumed to be piecewise continuously differentiable nonlinear mappings. The parameterized model for (1) can be represented as

$$\begin{aligned} \hat{x}_{k+1} &= \hat{f}(\hat{x}_k, u_k; \theta_f) + \hat{v}_k; \qquad \hat{x}(0) = \hat{x}_0 \\ \hat{y}_k &= \hat{g}(\hat{x}_k, u_k; \theta_g) + \hat{w}_k \end{aligned} \tag{2}$$

where $\hat{y} \in \mathbb{R}^l$ and $\hat{x} \in \mathbb{R}^n$ represent the output predictor vector, state predictor vector, θ_f and θ_g are vectors of model parameters to be estimated, $\hat{x}_0$ is the initial condition of the model's state, and $\hat{v} \in \mathbb{R}^n$ and $\hat{w} \in \mathbb{R}^l$ are modeled as Gaussian random processes with prior known means and standard deviations. The identification of the process model is the determination of θ_f and θ_g such that real process (1) is closely approximated by (2) (Kim et al., 2018). In practice, given the existence and uniqueness of solution for the dynamical system, the norm-bounded trajectories of the difference between (1) and (2) are used to represent the closeness

of the approximation (Kim et al., 2018). Three popular parameterized model structures used to approximate a nonlinear dynamical system are NSSM, GIOM, and DRNN. In the parameterization of the algebraic nonlinearity in each model structure, a FANN is used, and dynamic backpropagation algorithms are used in determining (aka training) the unknown model parameters (Narendra and Parthasarathy, 1990). The mathematical structures of the models are given below.

2.2.1. Neural state-space model (NSSM)

This state-space model uses FANNs to represent nonlinear mappings in the state and output equations (Suykens et al., 1995). The nonlinear mappings of equations are chosen to have one hidden layer, respectively, and the dynamic of the nonlinear system can be represented as

$$\begin{aligned}\hat{x}_{k+1} &= W_{AB}\tanh(V_A\hat{x}_k + V_B u_k + \beta_{AB}) + v_k\\ \hat{y}_k &= W_{CD}\tanh(V_C\hat{x}_k + V_D u_k + \beta_{CD}) + w_k\end{aligned} \tag{3}$$

where $\hat{y}_k \in \mathbb{R}^l$, $u \in \mathbb{R}^m$, and $\hat{x}_k \in \mathbb{R}^n$ are the output, input, and state; h_x and $h_y \in \mathbb{Z}^+$ are the number of neurons in the state and output equations, $V_A \in \mathbb{R}^{h_x\times n}$, $V_B \in \mathbb{R}^{h_x\times m}$, $W_{AB} \in \mathbb{R}^{n\times h_x}$, $\beta_{AB} \in \mathbb{R}^{h_x}$, $V_C \in \mathbb{R}^{h_y\times n}$, $V_D \in \mathbb{R}^{h_y\times m}$, $\beta_{CD} \in \mathbb{R}^{h_y}$, and $W_{CD} \in \mathbb{R}^{l\times h_y}$ are the parameters to be estimated in the FANNs. The generality and completeness of NSSM has been proved (Kim et al., 2018).

2.2.2. Global input-output model (GIOM)

The nonlinear input-output model structure

$$\hat{y}_{k+1} = \hat{g}(u_k, \ldots, u_{k-q+1}, \hat{y}_k, \ldots, \hat{y}_{k-r+1}) + v_k \tag{4}$$

uses past observations of input and output to predict future output (Levin and Narendra, 1995; Narendra and Parthasarathy, 1990), where q and r are the input and output time horizons, respectively. Replacing $\hat{g}$ with a neural network results in the GIOM

$$\hat{y}_{k+1} = W_A\tanh(V_A\hat{Y}_k^r + V_B U_{k-1}^{q-1} + V_C u_k + \beta) + v_k \tag{5}$$

where $\hat{Y}_k^r \triangleq [\hat{y}_k^\top, \ldots, \hat{y}_{k-r+1}^\top]^\top \in \mathbb{R}^{lr\times 1}$, $U_{k-1}^{q-1} \triangleq [u_{k-1}^\top, \ldots, u_{k-q+1}^\top]^\top \in \mathbb{R}^{m(q-1)\times 1}$, $W_A \in \mathbb{R}^{l\times h}$, $V_A \in \mathbb{R}^{h\times lr}$, $V_B \in \mathbb{R}^{h\times m(q-1)}$, $V_C \in \mathbb{R}^{h\times m}$, and $\beta \in \mathbb{R}^h$. The generality and completeness of GIOM has been proved (Levin and Narendra, 1995).

2.2.3. Dynamic recurrent neural network (DRNN)

The DRNN

$$\begin{aligned}\hat{x}_{k+1} &= -\alpha\hat{x}_k + W_{AB}\tanh(V_A\hat{x}_k + V_B u_k + \beta_{AB}) + v_k\\ \hat{y}_k &= W_{CD}\tanh(V_C\hat{x}_k + V_D u_k + \beta_{CD}) + w_k\end{aligned} \tag{6}$$

includes a recursive term in the state equation (Jin and Gupta, 1996; Fang and Kincaid, 1996), where α is the self-feedback parameter chosen to satisfy $0 < \alpha \leq 1$ and controls the incremental decay of state. The generality and completeness of DRNN has been proved (Jin and Gupta, 1996).

2.3. Standard nonlinear operator form (SNOF)

In modern robust linear control theory, the Linear Fractional Transformation (LFT) is a standard structure for representing linear systems with some forms of uncertainty. Some past publications (e.g., Suykens et al., 1995) have incorrectly stated that the NSSM can be represented as an LFT. In actuality, the NSSM and other DANNs can be written in the Standard Nonlinear Operator Form (SNOF), which is a nonlinear model structure in which a linear time-invariant dynamical system is connected to feedback with a bounded static nonlinear operator that has been studied since the early 1940s by Lur'e and Postnikov (1944). The nonlinearities γ_i are typically assumed to be continuous, which holds for the neurons used in neural networks. Nonlinearities γ_i that have been widely studied include the hyperbolic tangent and sigmoid. The nonlinearities can be nondifferentiable, such as a rectified linear unit (ReLU), which is a nonlinearity used in deep neural networks.

NSSM, GIOM, and DRNN can be represented in SNOF, indicating that the DANN models are subsets of the SNOF. The matrix $M = \begin{bmatrix} A & B \\ C & D \end{bmatrix}$ is a linear mapping between the inputs and outputs of the shift operator and the diagonal nonlinear operator Γ, where each diagonal element of Γ is the nonlinearity associated with a neuron in the network. The notation SNOF$(\frac{1}{z}I, M, \Gamma)$ can be used to represent the discrete-time equations

$$\begin{bmatrix} \hat{x}_{k+1} \\ q_k \\ \hat{y}_k \end{bmatrix} = \begin{bmatrix} A & B_p & B_u \\ C_q & D_{qp} & D_{qu} \\ C_y & D_{yp} & D_{yu} \end{bmatrix} \begin{bmatrix} \hat{x}_k \\ p_k \\ u_k \end{bmatrix}, \tag{7}$$

that is,

$$\hat{x}_{k+1} = A\hat{x}_k + B_p p_k + B_u u_k \tag{8}$$

$$q_k = C_q\hat{x}_k + D_{qp}p_k + D_{qu}u_k \tag{9}$$

$$\hat{y}_k = C_y\hat{x}_k + D_{yp}p_k + D_{yu}u_k \tag{10}$$

where the vectors q_k and p_k are the input and output of the nonlinear operator ($p_k = \Gamma(q_k)$), respectively, $A \in \mathbb{R}^{n\times n}$, $B_p \in \mathbb{R}^{n\times h}$, $B_u \in \mathbb{R}^{n\times m}$, $C_q \in \mathbb{R}^{h\times n}$, $D_{qp} \in \mathbb{R}^{h\times h}$, $D_{qu} \in \mathbb{R}^{h\times m}$, $C_y \in \mathbb{R}^{l\times n}$, $D_{yp} \in \mathbb{R}^{l\times h}$, $D_{yu} \in \mathbb{R}^{l\times m}$, and h is the total number of static nonlinearities. Equation (7) is similar with the form of state-space equation of a linear system. The major difference is that the SNOF represents the connection between the input and output of nonlinear activation functions.

When the nonlinearities γ_i are chosen to belong to the set of sector-bounded nonlinearities and/or monotonic static nonlinearities, the SNOF representation (7) can be expressed as a diagonal nonlinear differential inclusion or a Lur'e differential inclusion (Boyd et al., 1994). The SNOF is also very similar to the LFT, with the difference being that the operator Γ in the SNOF is a static nonlinear operator. For the structure where Γ is replaced by a norm-bounded operator, the (7) becomes the standard form for the design of robust linear controllers for discrete-time systems. Much of the same mathematical machinery applies to both model structures.

For $D_{qp} = 0$, the difference equations for the SNOF (7) are all explicit and solvable by first using (9) to compute q_k from $\hat{x}_k$ and u_k, then computing p_k from $p_k = \Gamma(q_k)$, and then inserting p_k into (8) and (10) to determine $\hat{x}_{k+1}$ and $\hat{y}_k$ respectively. The procedure for simulating the difference equations is the same for $D_{qp} \neq 0$, except for one difference, which is that the first step is to insert $\hat{x}_k$ and u_k into

$$q_k = C_q\hat{x}_k + D_{qp}\Gamma(q_k) + D_{qu}u_k \tag{11}$$

and numerically solve for q_k. There are several ways to solve this equation, including by zero-finding and fixed point iteration algorithms. One way is to numerically invert the equations off-line, i.e., define the function f by

$$f(q_k) := q_k - D_{qp}\Gamma(q_k) = C_q\hat{x}_k + D_{qu}u_k, \tag{12}$$

numerically solve for $f^{-1}(\cdot)$ off-line, and then apply on-line as

$$q_k = f^{-1}(C_q\hat{x}_k + D_{qu}u_k). \tag{13}$$

This off-line approach is most useful when D_{qp} is diagonal, as then the nonlinear operator on the left-hand side of (12) is diagonal, and the nonlinear inversions in (13) can be solved as independent scalar equations. For example, for the hyperbolic tangent activation function, each element of f is defined by

$$f_i(q_{k,i}) := q_{k,i} - D_{qp,ii}\tanh q_{k,i} \tag{14}$$

which are scalar nonlinearities where the $D_{qp,ii}$ are scalars. The nonlinear inverse f^{-1} is then the collection of scalar nonlinear inverses into a vector,

$$f^{-1} = [f_1^{-1}, \cdots, f_h^{-1}]^\top. \tag{15}$$

An advantage of setting D_{qp} to be zero during the training of a SNOF is that the on-line computational cost is lower when the model is used for real-time estimation and control.

2.3.1. Loop transformation

The nonlinear stability analysis tools for SNOFs exploit the fact that each nonlinearity γ_i lies within a sector. When the sector bounds assumed in a stability analysis condition does not match with the sector bounds in the SNOF, a loop transformation (Desoer and Vidyasagar, 1975) can be used to produce an equivalent SNOF.

2.3.2. Lur'e system stability

Consider a closed-loop system representable as a linear time-invariant (LTI) system interconnected with a static nonlinearity Γ:

$$\begin{aligned} \hat{x}_{k+1} &= A\hat{x}_k + Bp_k \\ q_k &= C\hat{x}_k \\ p_k &= \Gamma(q_k) \end{aligned} \tag{16}$$

This system is said to be *absolutely stable* if its equilibrium point at the origin is globally uniformly asymptotically stable for all memoryless nonlinearities in a given sector (Khalil and Grizzle, 2002; Vidyasagar, 2002).

The first absolute stability result was published based on the direct Lyapunov method (Lyapunov, 1892). A new form of Lyapunov function for nonlinear system was suggested by Lur'e and V.N. Postnikov (Lur'e and Postnikov, 1944), and numerous publications have employed this method (Liberzon, 2006). A different method to derive absolute stability conditions was proposed based on a frequency-domain approach (Popov, 1961). Thereafter, the generalized Kalman–Popov–Yakubovich lemma for multivariable systems was established (Gantmacher and Yakubovich, 1966; Popov, 1964). A popular quadratic criterion to study absolute stability was formulated (Yakubovich, 1998). The Lur'e system is a well-known benchmark problem, in which the system (16) satisfies: 1) (A, B, C) is a minimal realization, and 2) the nonlinear operator Γ: $\mathbb{R}^{n_q} \times \mathbb{Z}_+ \to \mathbb{R}^{n_q}$ is memoryless and sector-bounded, that is, lies within the sector-bound $\Phi_{sb}^{[\underline{\alpha},\overline{\alpha}]}$, i.e., $[\underline{\alpha}_i^{-1}\gamma_i(\sigma,k) - \sigma][\overline{\alpha}_i^{-1}\gamma_i(\sigma,k) - \sigma] \le 0$, $\forall \sigma \in \mathbb{R}$, $k \in \mathbb{Z}_+$, $i \in \{1, \ldots, n_q\}$ where n_q is the number of nonlinearities.

DANNs can be converted into a Lur'e system for which many sufficient conditions for global asymptotic stability conditions have been derived, e.g., see Kim et al. (2018) and citations therein. Much of the literature refers to the stability of the Lur'e system as *absolute stability* (Khalil and Grizzle, 2002). The next section presents a necessary condition for the stability of a Lur'e system, which is also a necessary stability condition when additional constraints are applied to the nonlinearities γ_i, such as being odd, monotonic, and locally slope restricted (Kim et al., 2018).

2.3.3. SNOF representation of DANNs

The SNOF representation of the DANNs can be used to test where every trajectory of the DANNs (NSSM, GIOM, or DRNN) converges to zero as $k \to \infty$. Γ is typically unit-sector-bounded with $\Gamma(0) = 0$, such as for the hyperbolic tangent and ReLU neurons, whereas the output of a sigmoid function does not vanish at the origin and is not sector-bounded. When the sigmoid function is chosen as an activation function, the loop transformation can be used to generate a SNOF with $\Gamma(0) = 0$ that is sector-bounded (Kim et al., 2018).

DANNs can be represented in SNOF, indicating that the SNOF is a superset of the DANNs. The relationships between the general values of the A, B, C in a SNOF to the particular values for the DANNs, derived using block diagram algebra, are given below.

- The NSSM can be written as

$$\begin{bmatrix} x_{k+1} \\ q_{x,k} \\ q_{y,k} \\ y_k \end{bmatrix} = \begin{bmatrix} 0 & W_{AB} & 0 & 0 \\ V_A & 0 & 0 & V_B \\ V_C & 0 & 0 & V_D \\ 0 & 0 & W_{CD} & 0 \end{bmatrix} \begin{bmatrix} x_k \\ p_{x,k} \\ p_{y,k} \\ u_k \end{bmatrix}. \tag{17}$$

  The A matrix is the zero matrix of $n \times n$ dimensions, $B_p = [W_{AB}\ 0_{n,h_y}]$, $B_u = 0_{n,m}$, $C_q = [V_A^\top V_C^\top]^\top$, $C_y = 0_{l,n}$, and $D_{qp} = 0_{h_x+h_y}$, $D_{qu} = [V_B^\top V_D^\top]^\top$, $D_{yp} = [0_{l,h_x}\ W_{CD}]$, $D_{yu} = 0_{l,m}$.
- The GIOM can be represented as $\hat{y}_{k+1} = W_A \tanh(q_k)$ where

$$\begin{aligned} q_k &= V_{A,1}\hat{y}_k + \cdots + V_{A,r}\hat{y}_{k-r+1} \\ &\quad + V_C u_k + V_{B,1}u_{k-1} + \cdots + V_{B,q-1}u_{k-q+1} \\ &= \left(\frac{1}{z}V_{A,1} + \frac{1}{z^2}V_{A,2} + \cdots + \frac{1}{z^r}V_{A,r}\right)W_A p_k \\ &\quad + \left(V_C + \frac{1}{z}V_{B,1} + \cdots + \frac{1}{z^{q-1}}V_{B,q-1}\right)u_k \end{aligned} \tag{18}$$

  where z^{-1} is the backward shift operator. For the general $r = q-1$ case, the minimal realization can be written as

$$\triangle = \begin{bmatrix} \frac{1}{z}I_h & & & & \\ & \frac{1}{z}I_h & & & \\ & & \ddots & & \\ & & & \frac{1}{z}I_h & \\ & & & & \frac{1}{z}I_h \end{bmatrix}_{rh\times rh},$$

$$A = \begin{bmatrix} 0_h & I_h & 0_h & \cdots & 0_h \\ 0_h & 0_h & I_h & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0_h \\ 0_h & \cdots & 0_h & 0_h & I_h \\ 0_h & \cdots & \cdots & 0_h & 0_h \end{bmatrix}_{rh\times rh},$$

$$B_p = \begin{bmatrix} V_{A,1}W_A \\ V_{A,2}W_A \\ \vdots \\ V_{A,r}W_A \end{bmatrix}_{rh\times h}, \quad B_u = \begin{bmatrix} V_{B,1} \\ V_{B,2} \\ \vdots \\ V_{B,q-1} \end{bmatrix}_{rh\times m},$$

$$C_q = [I_h\ 0_h\ \cdots\ 0_h]_{h\times rh},\ D_{qp} = 0_h,\ D_{qu} = V_C,$$
$$C_y = 0_{h\times rh},\ D_{yp} = W_A,\ D_{yu} = 0_{h,m}. \tag{19}$$

  In the case of $r \neq q-1$, the r in (19) is substituted by $r' = \max\{r, q-1\}$, and some elements (e.g., $V_{A,r}$ or $V_{B,q-1}$) are zero.
- The DRNN can be written as

$$\begin{bmatrix} x_{k+1} \\ q_{x,k} \\ q_{y,k} \\ y_k \end{bmatrix} = \begin{bmatrix} -\alpha I & W_{AB} & 0 & 0 \\ V_A & 0 & 0 & V_B \\ V_C & 0 & 0 & V_D \\ 0 & 0 & W_{CD} & 0 \end{bmatrix} \begin{bmatrix} x_k \\ p_{x,k} \\ p_{y,k} \\ u_k \end{bmatrix}. \tag{20}$$

  The A matrix is $-\alpha I_n$, and the other elements are same as for NSSM.

The above SNOF representation of GIOM is improved over that given in a previous study (Kim et al., 2018) which was not a minimal realization. The A matrix in the previous study was (21)

$$A = \begin{bmatrix} 0_{l,lr} & 0_l & 0_{l,m(q-1)} & 0_{l,m} \\ I_{lr} & 0_{lr,l} & 0_{lr,m(q-1)} & 0_{lr,m} \\ 0_{m,lr} & 0_{m,l} & 0_{m,m(q-1)} & 0_m \\ 0_{m(q-1),lr} & 0_{m(q-1),1} & I_{m(q-1)} & 0_{m(q-1),m} \end{bmatrix}_{l(r+1)+mq}. \tag{21}$$

The above expressions for DANNs imply that the SNOF can model or parameterize any nonlinear dynamical system with an arbitrarily small approximation error, just as for DANNs (Kim et al., 2018).

2.3.4. Controller form of the SNOF

The SNOF in (8)–(10) has excess degrees of freedom. This section shows how to employ a controller-form realization (Chen, 1998) to remove this excess, to simplify model identification.

The approach is illustrated for an example SNOF that has one input, one output, three states ($n = 3$), one neuron ($h = 1$), $D_{qp} = 0$, and $D_{yp} = 0$[1]:

$$A = \begin{bmatrix} -a_1 & -a_2 & -a_3 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \; B_p = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \tag{22}$$

$$C_q = \begin{bmatrix} c_{q1} & c_{q2} & c_{q3} \end{bmatrix}, \; B_u = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \tag{23}$$

$$C_y = \begin{bmatrix} c_{y1} & c_{y2} & c_{y3} \end{bmatrix}, \; D_{qu} = 0, \tag{24}$$

$$D_{yu} = \begin{bmatrix} d_{yu1} \end{bmatrix}, \; D_{qp} = 0, \; D_{yp} = 0. \tag{25}$$

This representation has an equivalent mapping between each variable, where the elements in the above equation can be derived from an associated transfer function. Searching for the real scalars in the above equations is less computationally expensive than for the general SNOF, especially for systems of high state dimension.

2.4. Realization, controllability, observability conditions

For convenience, this section summarizes some well-established control results used in this article.

2.4.1. Diagonalizable A matrix

For an A matrix with n distinct eigenvalues, the transfer function of a single-input single-output (SISO) system with n distinct poles can be represented as

$$\frac{b_1 z^{n-1} + \cdots + b_{n-1} z + b_n}{(z + p_1)(z + p_2) \cdots (z + p_n)}, \tag{26}$$

which can be rewritten by partial fraction expansion as

$$\frac{c_1}{z + p_1} + \frac{c_2}{z + p_2} + \cdots + \frac{c_n}{z + p_n}. \tag{27}$$

The realization of the transfer function into diagonal form can be represented as

$$A = \begin{bmatrix} -p_1 & & \\ & \ddots & \\ & & -p_n \end{bmatrix}, \; B = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}, \tag{28}$$

and

$$C = [c_1 \; \cdots \; c_n]. \tag{29}$$

Consider an $l \times m$ transfer function matrix for a multi-input multi-output (MIMO) system $G(z)$ in which every entry of the matrix is a proper coprime fraction. Let

$$D(z) = (z + p_1)(z + p_2) \cdots (z + p_{n-1})(z + p_n) \tag{30}$$

be the least common denominator of all entries of $G(z)$. Then $G(z)$ can be expressed as

$$\begin{aligned} G(z) &= \tfrac{1}{D(z)}[N(z)] \\ &= \tfrac{1}{D(z)}[N_1 z^{n-1} + N_2 z^{n-2} + \cdots + N_{n-1} z + N_n] \end{aligned} \tag{31}$$

where the N_i are $l \times m$ constant matrices. The $G(z)$ can be written as a partial fraction expansion,

$$G(z) = \frac{1}{z + p_1}\hat{N}_1 + \frac{1}{z + p_2}\hat{N}_2 + \cdots + \frac{1}{z + p_{n-1}}\hat{N}_{n-1} + \frac{1}{z + p_n}\hat{N}_n, \tag{32}$$

where the $\hat{N}_i$ and p_i are computed from the N_i and $D(z)$ using standard formulae (Chen, 1998).

Then a realization of $G(z)$ can be written as

$$A = I_m \otimes \begin{bmatrix} -p_1 & 0 & \cdots & 0 & 0 \\ 0 & -p_2 & 0 & \cdots & 0 \\ 0 & 0 & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ 0 & 0 & \cdots & 0 & -p_n \end{bmatrix}, \; B = I_m \otimes \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}, \tag{33}$$

and

$$C = \begin{bmatrix} \hat{N}_1 & \hat{N}_2 & \cdots & \hat{N}_{n-1} & \hat{N}_n \end{bmatrix}, \tag{34}$$

where I_m is the $m \times m$ unit matrix. The A matrix consists of n rows and n columns of $m \times m$ matrices, and the dimension of A matrix is $nm \times nm$. The dimension of B matrix is $nm \times m$. The C matrix consists of n number of $\hat{N}_i$, whose order is $l \times m$, and the dimension of C matrix is $l \times nm$.

Theorem 1. *(Theorem 6.D1 in Chen, 1998) The n-dimensional linear time-invariant state equation with (A, B, C) is controllable if and only if any of the following equivalent conditions are satisfied:*

(1) The $n \times nm$ controllability matrix

$$U \triangleq [B \vdots AB \vdots A^2B \vdots \cdots \vdots A^{n-1}B] \tag{35}$$

has rank n, where m is the number of inputs.

(2) For every eigenvalue λ_i of A, the $n \times (n + m)$ matrix $[\lambda_i I - A \vdots B]$ has rank n.

Theorem 2. *(Theorem 6.DO1 in Chen, 1998) The n-dimensional linear time-invariant state equation with (A, B, C) is observable if and only if any of the following equivalent conditions are satisfied:*

(1) The $nl \times n$ observability matrix

$$V \triangleq \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{n-1} \end{bmatrix} \tag{36}$$

has rank n, where l is the number of outputs.

(2) For every eigenvalue λ_i of A, the $(n + l) \times n$ matrix $\begin{bmatrix} \lambda_i I - A \\ C \end{bmatrix}$ has rank n.

2.4.2. Non-diagonalizable A matrix

For every square matrix A, there is a similarity transformation such that $\hat{A}$ has a Jordan normal form which features the eigenvalues with multiplicities collected in its diagonal elements and the superdiagonal elements being either 0 or 1 (sometimes in the subdiagonal elements instead of superdiagonal elements).

The Jordan form $\hat{A}$ with n states and q distinct eigenvalues with $r(i)$ geometric multiplicities can be written as

$$\hat{A} = \text{diag}(\hat{A}_1, \ldots, \hat{A}_q) \tag{37}$$

[1] The latter condition simplifies the algebra.

where $\hat{A}_i$ consists of all Jordan blocks associated with eigenvalue p_i, and

$$\hat{A}_i = \text{diag}(\hat{A}_{i1}, \ldots, \hat{A}_{ir(i)}). \tag{38}$$

The row of $\hat{B}$ corresponding to the last row of $\hat{B}_{ij}$ is denoted by b_{lij}, and the column of $\hat{C}$ corresponding to the first column of $\hat{C}_{ij}$ is denoted by c_{1ij}.

For an A matrix with $n = 5$ and the number of distinct eigenvalues $q = 3$ with multiplicities $r(i) = 2, 2, 1$, the rational transfer function of SISO system can be represented as

$$G(z) = \frac{b_1 z^4 + b_2 z^3 + b_3 z^2 + b_4 z + b_5}{(z+p_1)^2 (z+p_2)^2 (z+p_3)}, \tag{39}$$

which can be rewritten by partial fraction expansion as

$$\frac{c_1}{(z+p_1)^2} + \frac{c_2}{z+p_1} + \frac{c_3}{(z+p_2)^2} + \frac{c_4}{z+p_2} + \frac{c_5}{z+p_3}. \tag{40}$$

The realization of the transfer function into Jordan form can be represented as

$$A = \begin{bmatrix} -p_1 & 1 & 0 & 0 & 0 \\ 0 & -p_1 & 0 & 0 & 0 \\ 0 & 0 & -p_2 & 1 & 0 \\ 0 & 0 & 0 & -p_2 & 0 \\ 0 & 0 & 0 & 0 & -p_3 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \\ 1 \end{bmatrix}, \tag{41}$$

and

$$C = \begin{bmatrix} c_1 & c_2 & c_3 & c_4 & c_5 \end{bmatrix}.^{2} \tag{42}$$

Consider the 2×2 transfer matrix of multi-input multi-output (MIMO) system $G(z)$ in which every entry of the matrix is a proper coprime fraction. Let

$$D(z) = (z+p_1)^2 (z+p_2)^2 (z+p_3) \tag{43}$$

be the least common denominator of all entries of $G(z)$. Then $G(z)$ can be expressed as

$$G(z) = \begin{bmatrix} g_1(z) & g_2(z) \\ g_3(z) & g_4(z) \end{bmatrix} \tag{44}$$

where $g_i = \frac{c_{i1}}{(z+p_1)^2} + \frac{c_{i2}}{z+p_1} + \frac{c_{i3}}{(z+p_2)^2} + \frac{c_{i4}}{z+p_2} + \frac{c_{i5}}{z+p_3}$.

To derive the A, B, and C matrices that constitute M in the SNOF, construct a minimal realization of $G(z)$, via the Jordan canonical form which is

$$A = I \otimes \begin{bmatrix} -p_1 & 1 & 0 & 0 & 0 \\ 0 & -p_1 & 0 & 0 & 0 \\ 0 & 0 & -p_2 & 1 & 0 \\ 0 & 0 & 0 & -p_2 & 0 \\ 0 & 0 & 0 & 0 & -p_3 \end{bmatrix}, \quad B = I \otimes \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \\ 1 \end{bmatrix}, \tag{45}$$

and

$$C = \begin{bmatrix} c_{11} & c_{12} & c_{13} & c_{14} & c_{15} & c_{21} & c_{22} & c_{23} & c_{24} & c_{25} \\ c_{31} & c_{32} & c_{33} & c_{34} & c_{35} & c_{41} & c_{42} & c_{43} & c_{44} & c_{45} \end{bmatrix}. \tag{46}$$

Theorem 3 (Theorem 6.8 in Chen, 1998) The n-dimensional linear time-invariant Jordan-form dynamical system is controllable if and only if, for each $i = 1, 2, \ldots, q$, the rows of the $r(i) \times n$ matrix

$$B_i^l \triangleq \begin{bmatrix} b_{li1} \\ b_{li2} \\ \vdots \\ b_{lir(i)} \end{bmatrix} \tag{47}$$

are linearly independent. In addition, the system is observable if and only if, for each $i = 1, 2, \ldots, q$, the columns of the $n \times r(i)$ matrix

$$C_i^1 \triangleq [c_{1i1} \; c_{1i2} \; \cdots \; c_{1ir(i)}] \tag{48}$$

are linearly independent (over the field of complex numbers).

3. Matrix structure and eigenvalue analysis

In an analysis of a neural network structure, it is important to investigate how many parameters are optimized during the training procedure. In this context, the number of degrees of freedom of a neural network structure is defined as the maximum number of parameters (weights and biases) that should be fixed to have a completely distinguishable (distinct) neural network structure. When a connection between neurons in different hidden layers is represented as A, B, and C matrices, the size of matrices can be any size. Given that the uniqueness of the A, B, and C matrices is conserved for the permutation (Albertini et al., 1993), the matrices with the minimal condition (with both observability and controllability) are considered in this study. The number of independent elements in the minimal connection matrices is counted as the number of degrees of freedom of a neural network structure. When a neural network model has a larger number of degrees of freedom than the other model, then the former model can be represented as a superset of the latter model.

The first insights into limitations of the three DANN model structures is obtained by inspection of their A, B, and C matrices when written as a SNOF (see Section 2.3.3). In a SNOF, the A, B, and C matrices have no restrictions on their structure, as long as their elements are real. The situation is very different for the three DANNs; while the B and C matrices are allowed to have any structure, the DANNs have very restrictive structures for their A matrix. The A matrices are 0, $\begin{bmatrix} 0 & 0 \\ I & 0 \end{bmatrix}$, and $-\alpha I$ for the NSSM, GIOM, and DRNN model structures (Section 2.3.3). The eigenvalues of the first two DANNS are equal to zero, and for the latter DANN are all equal to $-\alpha$. These A matrices are highly restrictive compared to a SNOF, which can have any values for the eigenvalues including *complex*, and can have nondiagonal Jordan form. The very restrictive structure of the A matrices of the DANNs strongly suggests that much higher dimensionality would be needed in the DANNs to be able to fit the input-output behavior of nearly all nonlinear dynamical systems. This observation is consistent with the results of the case study reported by Kim et al. (2011), in which the SNOF captured the input-output behavior much more accurately than the DANNs when all of the model structures were restricted to have the same number of states and neurons.

The number of degrees of freedom in the A matrix is zero for the NSSM and GIOM and is one for the DRNN. The restriction of the A matrices and their eigenvalues for the DANNs are not removed by increasing the number of states. In contrast, the A matrix for the SNOF has degrees of freedom equal to the number of states in the model (i.e., its row or column dimension). This result can be proved by applying a nonsingular transformation matrix T to convert A to its Jordan form. To produce the same input-output mapping, the matrices B and C must be replaced by TB and CT^{-1}, respectively, which do not change the number of degrees of freedom. The maximum number of degrees of freedom for a Jordan

[2] The A matrix in (50) has five degrees of freedom. In (50), the degrees of freedom with respect to the values of the eigenvalues is three, and there are two degrees of freedom with respect to the superdiagonal elements since each 1 on the superdiagonal could alternatively be 0. As such, the A matrix for the SNOF has degrees of freedom equal to the number of states in the model irrespective of the number of distinct eigenvalues.

form is equal to the row/column dimension of the matrix A (any superdiagonal 1 adds a binary degree of freedom but is associated with a reduction in the number of distinct eigenvalues which removes a real degree of freedom).

4. Degree-of-Freedom analysis

This section derives the number of degrees of freedom for the SNOF and DANNs, using simple examples to illustrate the main idea of the proofs while saving space. To represent the characteristic of applications in having the same poles appear in multiple elements, the examples are parameterized in terms of s, which is the number of distinct poles that can appear in the elements of the transfer function $G(z) = C(zI - A)^{-1}B$ where the matrices are defined in (7).

The state equation of the general SNOF (8) can be written as

$$z\hat{x}_k = A\hat{x}_k + B_p p_k + B_u u_k \tag{49}$$

$$\hat{x}_k = (zI - A)^{-1}[B_p p_k + B_u u_k]. \tag{50}$$

Insertion of (50) into (9) and (10) results in

$$q_k = C_q(zI - A)^{-1}[B_p p_k + B_u u_k] + D_{qp} p_k + D_{qu} u_k \tag{51}$$

$$= [C_q(zI - A)^{-1}B_p + D_{qp}]p_k + [C_q(zI - A)^{-1}B_u + D_{qu}]u_k \tag{52}$$

$$= G_{qp} p_k + G_{qu} u_k \tag{53}$$

$$\hat{y}_k = C_y(zI - A)^{-1}[B_p p_k + B_u u_k] + D_{yp} p_k + D_{yu} u_k \tag{54}$$

$$= [C_y(zI - A)^{-1}B_p + D_{yp}]p_k + [C_y(zI - A)^{-1}B_u + D_{yu}]u_k \tag{55}$$

$$= G_{yp} p_k + G_{yu} u_k. \tag{56}$$

Then, the total transfer matrix can be represented with $G = \begin{bmatrix} G_{qp} & G_{qu} \\ G_{yp} & G_{yu} \end{bmatrix}$ where

$$\begin{aligned} G_{qp} &= C_q(zI - A)^{-1}B_p + D_{qp} \\ G_{qu} &= C_q(zI - A)^{-1}B_u + D_{qu} \\ G_{yp} &= C_y(zI - A)^{-1}B_p + D_{yp} \\ G_{yu} &= C_y(zI - A)^{-1}B_u + D_{yu}. \end{aligned} \tag{57}$$

The number of degrees of freedom is first analyzed for a SNOF in which each element can have up to $s = 2$ distinct poles and $h = 3$ neurons to establish a pattern that is used to give the expression in the general case.

4.1. SNOF With distinct eigenvalues

For $s = 2$, $h = 3$, consider one element of the general transfer function matrix G_{qp} can be rewritten as $G_{qp}(z) = \frac{1}{D(z)} N_{qp}(z)$ where

$$D(z) = (z + p_1)(z + p_2), \tag{58}$$

$$N_{qp}(z) = \begin{bmatrix} a_1(z + b_1) & a_2(z + b_2) & a_3(z + b_3) \\ a_4(z + b_4) & a_5(z + b_5) & a_6(z + b_6) \\ a_7(z + b_7) & a_8(z + b_8) & a_9(z + b_9) \end{bmatrix}, \tag{59}$$

which by partial fraction expansion can be written as

$$G_{qp}(z) = \begin{bmatrix} \frac{c_1}{z+p_1} + \frac{d_1}{z+p_2} & \frac{c_2}{z+p_1} + \frac{d_2}{z+p_2} & \frac{c_3}{z+p_1} + \frac{d_3}{z+p_2} \\ \frac{c_4}{z+p_1} + \frac{d_4}{z+p_2} & \frac{c_5}{z+p_1} + \frac{d_5}{z+p_2} & \frac{c_6}{z+p_1} + \frac{d_6}{z+p_2} \\ \frac{c_7}{z+p_1} + \frac{d_7}{z+p_2} & \frac{c_8}{z+p_1} + \frac{d_8}{z+p_2} & \frac{c_9}{z+p_1} + \frac{d_9}{z+p_2} \end{bmatrix}. \tag{60}$$

Both of the above transfer matrices have $sh^2 + s$ degrees of freedom since the p_i, a_j, and b_k cannot be changed in any way without changing the input-output relationship for $G_{qp}(z)$. In the case of the other elements (G_{qp}, G_{qu}, and G_{yp}), the matrices have same poles (p_1 and p_2), but have independent elements in $N_{qu} \in \mathbb{R}^{h \times m}$, $N_{yp} \in \mathbb{R}^{l \times h}$, and $N_{yu} \in \mathbb{R}^{l \times m}$. Therefore, the total number of degrees of freedom in the final transfer matrix $G = \begin{bmatrix} G_{qp} & G_{qu} \\ G_{yp} & G_{yu} \end{bmatrix}$ is $s(h^2 + hm + hl + lm + 1)$

To characterize the set of A, B, C matrices in the SNOF that corresponds to $G(z)$, write the realization of the transfer function matrix (60) as a diagonal state-space model,

$$\hat{A} = T^{-1}AT = \begin{bmatrix} -p_1 I & 0 \\ 0 & -p_2 I \end{bmatrix}, \quad \hat{B} = T^{-1}B = \begin{bmatrix} I \\ I \end{bmatrix}, \tag{61}$$

and

$$\hat{C} = CT = \begin{bmatrix} c_1 & c_2 & c_3 & d_1 & d_2 & d_3 \\ c_4 & c_5 & c_6 & d_4 & d_5 & d_6 \\ c_7 & c_8 & c_9 & d_7 & d_8 & d_9 \end{bmatrix}, \tag{62}$$

where each identity matrix $I \in \mathbb{R}^{3\times3}$, each zero matrix $0 \in \mathbb{R}^{3\times3}$, and $T \in \mathbb{R}^{3\times3}$ is any invertible matrix that diagonalizes A.

This state-space model has the same $sh^2 + s$ degrees of freedom. The minimality of this realization of (A, B, C) can be confirmed by analyzing the controllability and observability of the system. Controllability is shown by noting that

$$V = [\hat{B}\ \hat{A}\hat{B}] = \begin{bmatrix} I & -p_1 I \\ I & -p_2 I \end{bmatrix} \tag{63}$$

has full rank for $p_1 \neq p_2$.

Observability holds if $\begin{bmatrix} \lambda_i I - \hat{A} \\ \hat{C} \end{bmatrix}$ has the same rank for all of its eigenvalues λ_i. Since

$$\lambda_i I - \hat{A} = \mathrm{diag}(\lambda_i + p_1, \lambda_i + p_1, \lambda_i + p_1, \lambda_i + p_2, \lambda_i + p_2, \lambda_i + p_2), \tag{64}$$

the matrix $\lambda_i I - \hat{A}$ has a rank of three since λ_i is equal to $-p_1$ or $-p_2$, and the matrix $\hat{C}$ can always choose its first three columns and its second three columns to be full rank by minor perturbations of c_i and d_j. In other words, $\hat{C}$ can always be perturbed by an arbitrarily small amount so that the state-space model is observable, while fitting $G(z)$ arbitrarily closely.

4.2. SNOF For repeated poles

Consider a system with $q < s$ distinct poles p_i $(1 \le i \le q)$ of algebraic multiplicity m_i, then the jth entry of the transfer function matrix associated with the SNOF can be represented as

$$g_j(z) = \frac{b_{j1}z^{s-1} + \cdots + b_{js-1}z + b_{js}}{(z + p_1)^{m_1} \ldots (z + p_q)^{m_q}}, \tag{65}$$

which can be divided into q subsystems

$$g_j(z) = g_{j1}(z) + \cdots + g_{jq}(z), \tag{66}$$

each of which can be written by partial fraction expansion as

$$g_{ji}(z) = \frac{c_{j,1i}}{z + p_i} + \cdots + \frac{c_{j,m_i i}}{(z + p_i)^{m_i}}. \tag{67}$$

Let the number of Jordan blocks associated with pole p_i be r_i (its geometric multiplicity); then $r_i \leq h$ for all i to have controllability and observability.

For up to $s = 5$ states per element and $h = 2$ neurons, consider the system with the transfer function matrix,

$$G_{qp}(z) = \frac{1}{D(z)}N(z) = \frac{1}{D(z)}\begin{bmatrix} n_1(z) & n_2(z) \\ n_3(z) & n_4(z) \end{bmatrix}, \tag{68}$$

where $D(z) = (z+p_1)^2(z+p_2)^2(z+p_3)$ and

$$n_j(z) = b_{j1}z^4 + b_{j2}z^3 + b_{j3}z^2 + b_{j4}z + b_{j5} \tag{69}$$

for $j = 1, \ldots, 4$. The number of real degrees of freedom for this example is $2^2 5 + 3$, which are the values of the b_{ji} and p_i.

Extrapolating the expressions to general s, h, and q, the number of real degrees of freedom for G_{qp} is $sh^2 + q$. Effectively the example has two integer degrees of freedom, which is the number of poles that are repeated, so the *total* number of degrees of freedom for G_{qp} is $sh^2 + s$.

The realization of (68) can be represented as (45) and (46), and the construction also has $sh^2 + q$ real degrees of freedom. In the case of the other elements (G_{qp}, G_{qu}, and G_{yp}), the matrices have same poles (p_1, p_2, and p_3), but have independent elements in N_{qu}, N_{yp}, and N_{yu} (similar to distinct pole case). Then, final G matrix has $s(h^2 + hm + hl + lm) + q$ degrees of freedom.

To show controllability of the Jordan form, the last rows of k_{i1} and k_{i2} are linearly independent, respectively, for all distinct poles p_i (Theorem 3). For p_1, the last row of k_{11} is [1 0] and of k_{12} is [0 1], which are linearly independent. This also holds for p_2. For p_3, the last row of k_{31} is [0 1], which implies that the system is controllable.

The analysis is similar for showing observability.

4.3. Degrees of freedom for the three DANNs

4.3.1. NSSM

The NSSM is the subset of SNOF in which all eigenvalue are zero ($p_1 = \cdots = p_s = 0$). For example, for an NSSM with up to $s = 1$ state per element and $h = 2$ neurons, the transfer function matrix is

$$G_{qp}(z) = \begin{bmatrix} \frac{c_1}{z} & \frac{c_2}{z} \\ \frac{c_3}{z} & \frac{c_4}{z} \end{bmatrix}. \tag{70}$$

The realization of transfer function matrix into diagonal form gives

$$A = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}, \; B_p = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \; C_q = \begin{bmatrix} c_1 & c_2 \\ c_3 & c_4 \end{bmatrix}. \tag{71}$$

The controllability matrix is full rank and the observability matrix is full rank provided that the matrix C_q is full rank. The set of matrices that is full rank is dense in the space of 2×2 real matrices.

Both the transfer function matrix and A, B_p, C_q matrices have only $h^2 = 4$ degrees of freedom. In general the number of degrees of freedom for G_{qp} is sh^2. For NSSM, from $B_u = 0$, $C_y = 0$, $D_{qp} = 0$, and $D_{yu} = 0$, (57) is reduced into $G_{qu} = D_{qu} = [V_B^\top \; V_D^\top]^\top$, $G_{yp} = D_{yp} = [0 \; W_{CD}]$, $G_{yu} = 0$. Therefore, the transfer matrix G has $h^2 + 2hm + hl$ ($V_B, V_D \in \mathbb{R}^{h \times m}$ and $W_{CD} \in \mathbb{R}^{l \times h}$).

4.3.2. GIOM

The GIOM can be represented as a subset of the general case. For a GIOM with up to $s = 2 (r = q - 1 = 2)$ states per element, $h = 3$ neurons, and $m = 3$, the transfer function matrix $G_{qp}(z)$ of GIOM can be written as

$$G_{qp}(z) = \begin{bmatrix} \frac{a_1}{z^2} + \frac{b_1}{z} & \frac{a_2}{z^2} + \frac{b_2}{z} & \frac{a_3}{z^2} + \frac{b_3}{z} \\ \frac{a_4}{z^2} + \frac{b_4}{z} & \frac{a_5}{z^2} + \frac{b_5}{z} & \frac{a_6}{z^2} + \frac{b_6}{z} \\ \frac{a_7}{z^2} + \frac{b_7}{z} & \frac{a_8}{z^2} + \frac{b_8}{z} & \frac{a_9}{z^2} + \frac{b_9}{z} \end{bmatrix}, \tag{72}$$

Table 1
The number of degrees of freedom in the DANNs and SNOF.

Model Structure	# of degrees of freedom
SNOF (distinct)	$s(h^2 + hm + hl + lm + 1)$
SNOF (general)	$s(h^2 + sm + hl + lm) + q$
NSSM	$h^2 + 2hm + hl$
GIOM	$s(h^2 + hm) + hl$
DRNN	$sh^2 + 2hm + hl + 1$

with state-space matrices given by

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \; B_p = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{73}$$

and

$$C_q = \begin{bmatrix} a_1 & b_1 & a_2 & b_2 & a_3 & b_3 \\ a_4 & b_4 & a_5 & b_5 & a_6 & b_6 \\ a_7 & b_7 & a_8 & b_8 & a_9 & b_9 \end{bmatrix}. \tag{74}$$

The nonsingularity of the controllability matrix implies that (A, B_p) is controllable, and the observability matrix is nonsingular for almost all choices of degrees of freedom.

In both of the above representations, the number of degrees of freedom for G_{qp} is $3^2 2 = 18$ for the example and sh^2 in general. From (57),

$$G_{qu}(z) = \begin{bmatrix} \frac{a'_1}{z^2} + \frac{b'_1}{z} + c'_1 & \frac{a'_2}{z^2} + \frac{b'_2}{z} + c'_2 & \frac{a'_3}{z^2} + \frac{b'_3}{z} + c'_3 \\ \frac{a'_4}{z^2} + \frac{b'_4}{z} + c'_4 & \frac{a'_5}{z^2} + \frac{b'_5}{z} + c'_5 & \frac{a'_6}{z^2} + \frac{b'_6}{z} + c'_6 \\ \frac{a'_7}{z^2} + \frac{b'_7}{z} + c'_7 & \frac{a'_8}{z^2} + \frac{b'_8}{z} + c'_8 & \frac{a'_9}{z^2} + \frac{b'_9}{z} + c'_9 \end{bmatrix}, \tag{75}$$

$G_{yp} = D_{yp} = W_A$, and $G_{yu} = 0$. Thus, G_{qu} and G_{yp} have shm and hl degrees of freedom, respectively, and the final G matrix have $sh^2 + shm + hl$ degrees of freedom.

4.3.3. DRNN

DRNN is the subset of SNOF in which all eigenvalue are same ($p_1 = \cdots = p_s$). For up to $s = 1$ state per element and $h = 2$ neurons, the corresponding transfer function matrix is

$$G_{qp}(z) = \begin{bmatrix} \frac{c_1}{z+p_1} & \frac{c_2}{z+p_1} \\ \frac{c_3}{z+p_1} & \frac{c_4}{z+p_1} \end{bmatrix}. \tag{76}$$

The realization of transfer function matrix into diagonal form can be written as follows:

$$A = \begin{bmatrix} -p_1 & 0 \\ 0 & -p_1 \end{bmatrix}, \; B_p = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \; C_q = \begin{bmatrix} c_1 & c_2 \\ c_3 & c_4 \end{bmatrix}. \tag{77}$$

The system is controllable, and is observable for any full-rank matrix C_q. The number of degrees of freedom of A is 1 from p_1, and the total number of degree of freedom of (A, B_p, C_q) is $1 + 2^2 = 5$. For a DRNN with up to s states per element and h neurons, the total number of degrees of freedom of (A, B_p, C_q) is $sh^2 + 1$. For DRNN, from $B_u = 0$, $C_y = 0$, $D_{qp} = 0$, and $D_{yu} = 0$, (57) is reduced into $G_{qu} = D_{qu} = [V_B^\top \; V_D^\top]^\top$, $G_{yp} = D_{yp} = [0 \; W_{CD}]$, $G_{yu} = 0$. Therefore, the transfer matrix G has $sh^2 + 2hm + hl + 1$ ($V_B, V_D \in \mathbb{R}^{h \times m}$ and $W_{CD} \in \mathbb{R}^{l \times h}$).

4.3.4. Summary

Table 1 summarizes the number of degrees of freedom for all of the model structures. For any fixed number of neurons, the SNOF has a larger number of degrees of freedom that can be optimized to fit input-output data of the nonlinear dynamical system during model identification. The additional degrees of freedom suggest that, for systems with the relatively low number of neurons

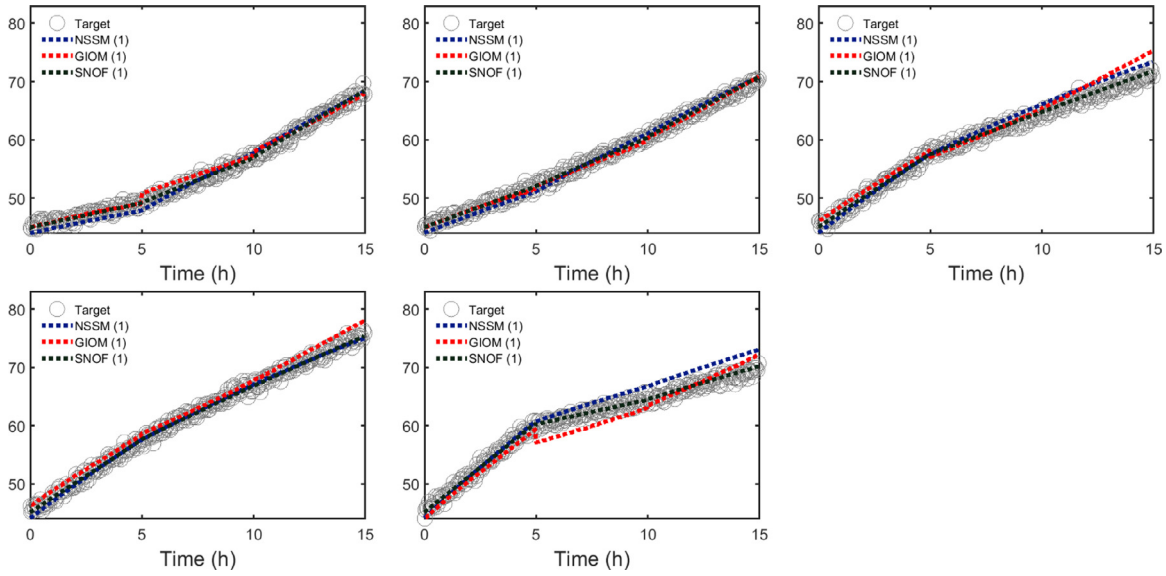

Fig. 1. NSSM, GIOM, SNOF predictions for the biomass concentration (g/L) for five test operating conditions. The number in parenthesis in the legend is the number of hidden neurons. The figures in the first row denote test sets 1–3, and in the second row denote test sets 4 and 5.

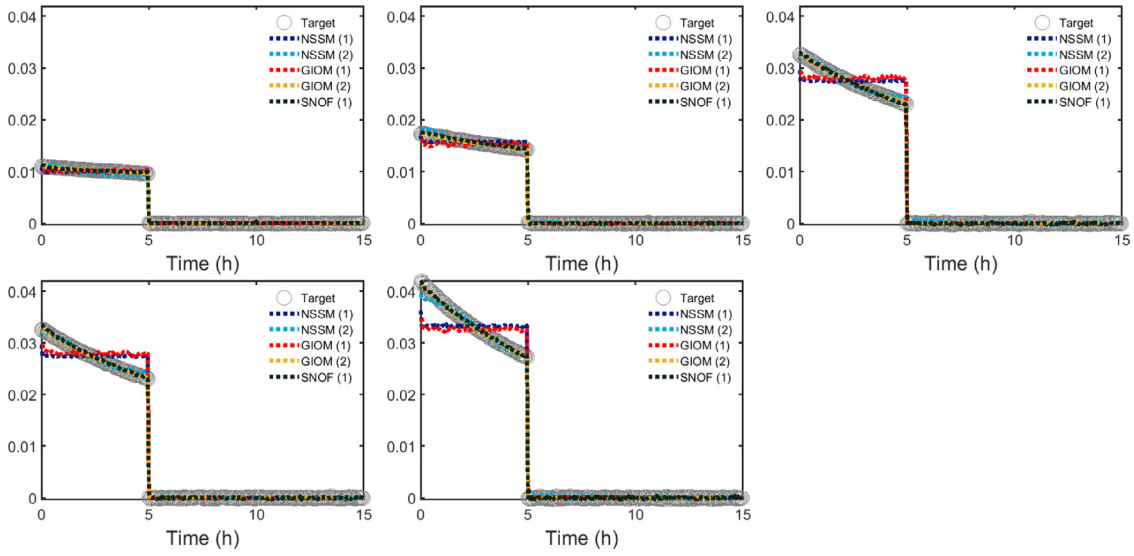

Fig. 2. NSSM, GIOM, and SNOF predictions for the glycerol concentration (g/L) for five test operating conditions. The number in parenthesis in the legend implies the number of hidden neurons. The figures in the first row denote test sets 1–3, and those in the second row denote test sets 4 and 5.

needed to have nonconservative analysis, an accurate SNOF model has the potential to have fewer neurons than the three popular DANNs. For example, a SNOF with $s = 5$ and *one* neuron $h = 1$ would have more degrees of freedom ($sh^2 + s = 10$) than an NSSM or GIOM with one state and *three* neurons ($sh^2 = 9$). These results motivate the further exploration of methods for the optimal training of SNOFs and their application to specific processes, to demonstrate the expected reduction in conservatism when applying rigorous analysis and synthesis methods compared to DANNs.

5. Case study

This section compare the performance and compactness of the DANNs and SNOF for a system with highly nonlinear dynamics.

5.1. Multistage bioreactor model

Most biochemical processes have nonlinear kinetics. *Pichia pastoris* is used to manufacture many proteins, due in part to its fast growth and in part to its easy regulation (Ren et al., 2003). A mechanistic model describing the growth and energy metabolism of *Pichia pastoris* cells producing protein in a bioreactor with various operation modes serves as the "true process" for comparing the size of DANNs and SNOF needed to model the input-output behavior of the process. The process consists of two stages. At first, the cells are fed glycerol to grow biomass in fed-batch mode with a step glycerol feed. The second stage is methanol growth in fed-batch mode with two methanol feed steps, in which feed rates are increasing in the second step. In this study, a macroscopic bioreactor model for the process is used to produce datasets, which are used in identification (i.e., training) and testing the DANNs and SNOF models. The differential equations for the mass balances for the system are

$$\frac{dX}{dt} = \frac{F_{in}}{V}(X_{in} - X) + (\mu_g + \mu_m)X \tag{78}$$

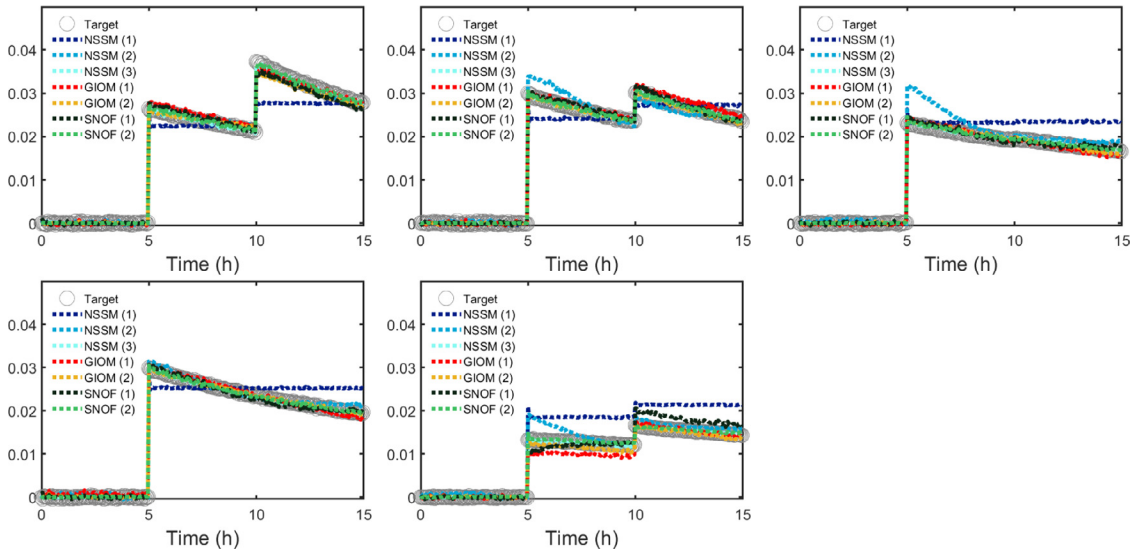

Fig. 3. NSSM, GIOM, and SNOF predictions for the methanol concentration (g/L) for five test operating conditions. The number in parenthesis in the legend implies the number of hidden neurons. The figures in the first row denote test sets 1–3, and those in the second row denote test sets 4 and 5.

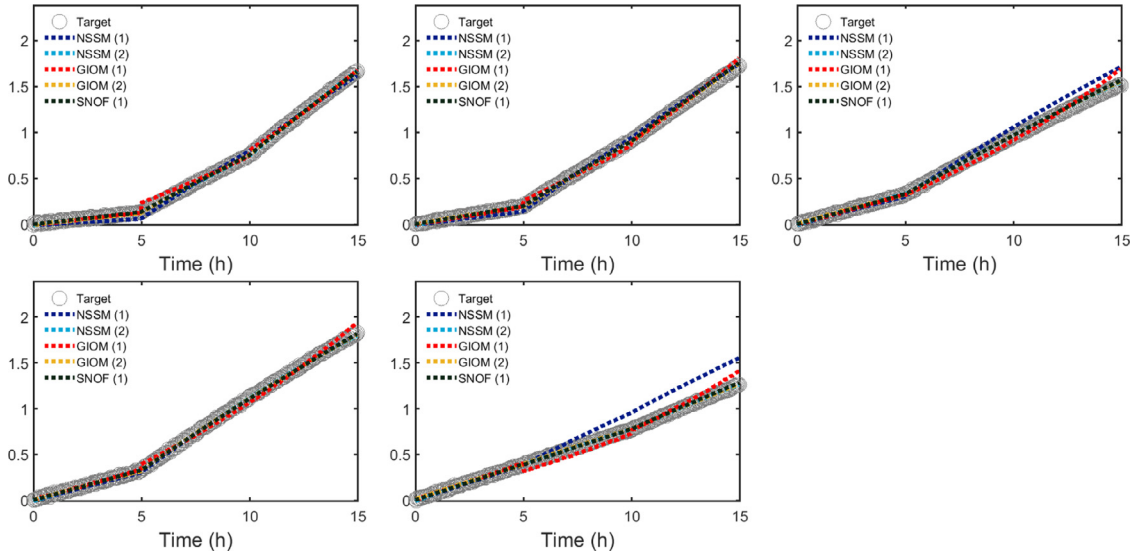

Fig. 4. NSSM, GIOM, and SNOF predictions for the product concentration (g/L) for five test operating conditions. The number in parenthesis in the legend implies the number of hidden neurons. The figures in the first row denote test set 1–3, and those in the second row denote test sets 4 and 5.

$$\frac{dS_g}{dt} = \frac{F_{in}}{V}(S_{in,g} - S_g) - q_{s,g}X \tag{79}$$

$$\frac{dS_m}{dt} = \frac{F_{in}}{V}(S_{in,m} - S_m) - q_{s,m}X \tag{80}$$

$$\frac{dP}{dt} = -\frac{F_{in}}{V}P + q_P X \tag{81}$$

where X is the concentration of biomass from dry weight (g/L), F is the glycerol or methanol feed rate (mL/h) in which subscripts *in* and *out* denote inlet and outlet respectively, V is the medium volume (L), μ_g is the specific growth rate (1/h), S_{in} is the inlet substrate concentration (g/L), S is the limiting substrate concentration, q_s is the specific rate of substrate consumption (g/g-h), P is the concentration of product (g/L), q_P is the specific rate of product generation (g/g-h), and subscripts g and m denote glycerol and methanol, respectively.

The specific rate of substrate consumption $q_{s,i}$ (g substrate/g biomass-h) in the substrate limitation regime was assumed to follow Monod kinetics (Monod, 1949),

$$q_{s,i} = q_{s,i}^{max}\frac{S_i}{S_i + K_{S,i}}, \tag{82}$$

where $q_{s,i}^{\max}$ is the specific maximum rate of substrate consumption (g substrate/g biomass-h) and $K_{S,i}$ is the saturation constant (g/L).

The specific growth rate at the limiting substrate μ_i (1/h) was calculated from the difference between the specific rate of substrate consumption and maintenance multiplied by the yield coefficient,

$$\mu_i = (q_{s,i} - q_{m,i})Y_{em,i}, \tag{83}$$

where $q_{m,i}$ is the maintenance coefficient (g/g-h) and $Y_{em,i}$ is the biomass yield coefficient exclusive maintenance (g/g). The specific production rate q_P (g/g-h) was assumed to follow a specific enzyme model (Hong et al., 2021). The responsible enzyme for protein secretion is assumed to be only active when the enzyme E is

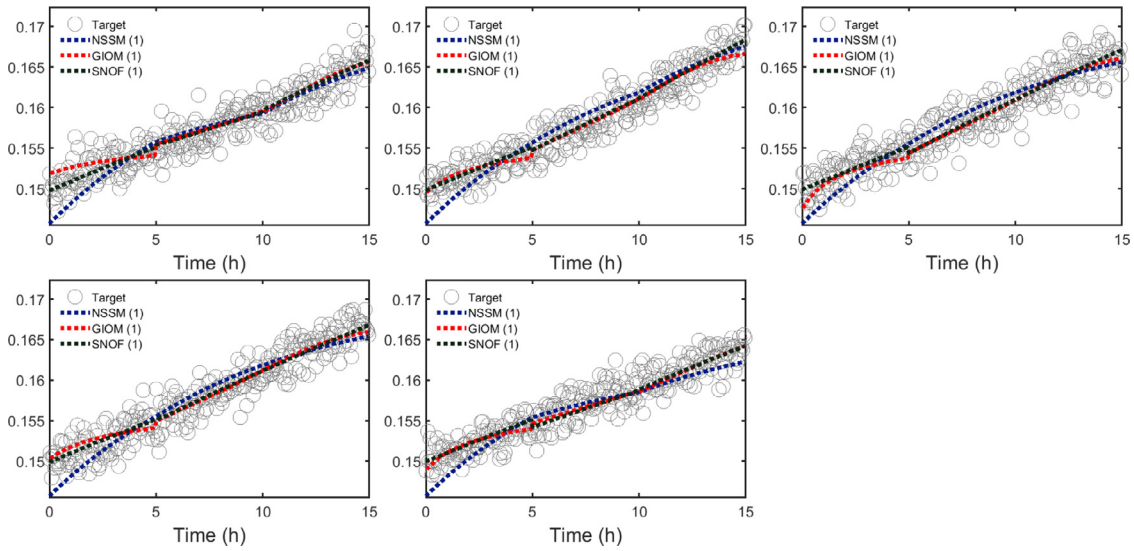

Fig. 5. NSSM, GIOM, and SNOF predictions for the medium volume (L) for five test operating conditions. The number in parenthesis in the legend implies the number of hidden neurons. The figures in the first row denote test sets 1–3, and those in the second row denote test sets 4 and 5.

Table 2
The parameters used in the simulation.

Symbol	Unit	Glycerol	Methanol
$K_{S,i}$	g/L	0.1	0.1
$q_{m,i}$	g/g-h	0	0.013
$q_{s,i}^{\max}$	g/g-h	0.37	0.57
$S_{in,i}$	g/L	250–750	780–1170
$Y_{em,i}$	g/g	0.7	0.36
$F_{in,i}$	mL/h	1	0.7–1.5
$pK_{a,E}$		3.42	
α_P	mg/g	23.2	69.0

Table 3
The operational modes.

Operation Mode		1	2	3
Feed inlet (mL/h)		$F_{in,g}$	$F_{in,m}^1$	$F_{in,m}^2$
Feed inlet	Glycerol	$S_{in,g}$	0	0
Concentration (g/L)	Methanol	0	$S_{in,m}$	$S_{in,m}$
Operation Time (h)		5	5	5

deprotonated: $EH^+ \overset{K_{a,E}}{\rightleftarrows} E + H^+$, where $K_{a,E}$ is the acid dissociation constant for the enzyme. Then the specific total production rate of the active enzyme is

$$q_{P,i} = \frac{\alpha_{P,i}\mu_i}{10^{pK_{a,E}-pH}+1} \tag{84}$$

where $\alpha_{P,i}$ is the growth-associated proportional coefficient. The parameters of the model are obtained in the literature (Hong et al., 2021) and shown in Table 2. The pH of system was fixed to 6.5, and the precipitation reaction in the medium and the intracellular changes in carbon and nitrogen concentrations in protein were assumed to be negligible.

This particular bioreactor operation was selected because of its increased nonlinear dynamics associated with the discrete jump in its model parameters when transitioning between a glycerol and methanol feed.

5.2. Global optimization with multistart

For the identification procedure (i.e., network training), multiple algorithms are available for finding the weights and biases to minimize the error function, which is typically the sum-of-squared differences between the model predictions and data. Gradient-based algorithms can converge quickly, but usually only to local minima. So-called "global optimization algorithms" such as evolutionary algorithms have been applied to network training, which can converge to single or multiple global minima (Volgis and Lagaris, 2006). Such global minima may have the same fit to data for the data used for model evaluation, but may produce different predictions or interpolations when fed different inputs.

The multistart method, which runs optimization algorithms for many random choices for initial guesses, is widely applied in parameter estimation. When a large enough number of well-distributed starting points are used, the method can find the basin of attraction which includes the global optima (Peri and Tinti, 2012). In this study, the training of DANNs and SNOF was repeated until eight initial guesses converged to the same minimum error, with thousands of starting points made randomly in

Table 4
The operating conditions for the training and test datasets ($F_{in,g} = 1$ mL/h).

Data set	No.	$S_{in,g}$ (g/L)	$F_{in,m}$ (mL/h)		$S_{in,m}$ (g/L)
			$F_{in,m}^1$	$F_{in,m}^2$	
Training	1	250	1.1848	1.2867	780.0
	2	250	0.8112	0.9501	877.5
	3	250	0.8025	1.4978	975.0
	4	250	1.2297	1.3274	1170.0
	5	375	0.7469	0.7775	780.0
	6	375	0.7693	1.1490	975.0
	7	375	1.1932	1.4711	1072.5
	8	375	0.9969	1.1594	1170.0
	9	500	0.8615	1.0617	780.0
	10	500	0.8561	1.1554	877.5
	11	500	1.0811	1.1670	975.0
	12	500	1.1143	1.3585	1072.5
	13	500	0.7552	1.2858	1170.0
	14	625	0.8636	1.3624	877.5
	15	625	0.9409	1.3974	1072.5
	16	625	1.1717	1.4854	1170.0
	17	750	1.0454	1.1071	877.5
	18	750	1.2672	1.4328	975.0
	19	750	1.1711	1.4125	1072.5
	20	750	0.9738	1.2095	1170.0
Test	1	250	0.8432	1.3023	1072.5
	2	375	1.1932	1.4699	877.5
	3	625	1.2148	1.2377	780.0
	4	625	1.1940	1.1942	975.0
	5	750	0.8011	1.0538	780.0

Table 5
Comparison of performances of SNOF and DANNs.

	Averaged Relative Error (%)							
	h	n	SNOF	NSSM	h	q	r	GIOM
Biomass concentration (g/L)	1	1	1.01	2.00	1	2	1	2.10
Glycerol concentration (g/L)	1	1	2.83	15.12	1	2	1	15.36
	2	2		5.58	2	2	1	1.50
Methanol concentration (g/L)	1	1	7.14	24.79	1	2	1	7.61
	2	2	2.00	12.44	2	2	1	4.31
	3	3		2.10				
Product concentration (g/L)	1	1	1.85	10.5	1	2	1	6.24
	2	2		1.78	2	2	1	1.79
Medium volume (L)	1	1	1.00	1.30	1	2	1	1.04

the range of $[-1, 1]$ to learn the best weights to predict the test set well.

5.3. Model identification

5.3.1. Simulation details

Datasets were produced by running twenty-five simulations for a total process time of 15 h, with a sampling period of 3 minutes for on-line measurements of four input variables ($F_{in,g}$, $F_{in,m}$, $S_{in,g}$, and $S_{in,m}$) and five output variables (X, S_g, S_m, P, and V), respectively. The detailed operational modes of the multistage bioreactor are listed in Table 3.

The initial states of bioreactors are fixed to the same for all simulations: $X_0 = 45$ g/L, $S_{g0} = S_{m0} = P_0 = 0$ g/L, and $V_0 = 0.15$ L. The simulations provide a wide variation in the datasets by varying feed rates and concentrations.

For the first stage (semi-batch glycerol feed), the initial feed rate of glycerol ($F_{in,g}$) was fixed to 1 mL/h while the initial concentration of the glycerol in feed ($S_{in,g}$) ranged from 250 to 750 g/L. For the second stage (semi-batch on methanol feed), the first and second feed rate of methanol ($F^1_{in,m}$ and $F^2_{in,m}$) ranged from 0.7 to 1.5 mL/h while the initial concentration of the methanol in feed ($S_{in,m}$) ranged from 780 to 1170 g/L. All bioreactors were simulated with additive white Gaussian noise ($\sigma = 0.1$) in the inputs and outputs.

5.3.2. Training procedure

After the simulations, twenty datasets were randomly allocated to the training set while the remaining datasets were allocated to test set. The detailed conditions of resulted training and test set at each operational mode are given in Table 4.

The minimum and maximum values of input variables and output variable in the training set were used to normalize all data into the range of $[0, 1]$. In order to find out the required number of neurons to capture the dynamics of each output, the DANNs and SNOF were trained in the form of four inputs and one output, respectively. The training was conducted with back-propagation through time that minimizes the Mean Squared Error (MSE) between target and corresponding one-step-head prediction of the training set. The DAANs and SNOF were trained with the multiple initial guesses, and the prediction accuracy of the trained DANNs and SNOF was quantified in terms of MSE when applying the model to the test set. The performance of DANNs and SNOF was estimated after the re-scaling of the prediction into the original range. In order to compare the relative performance on output variables, the averaged relative error for j output (ARE_j) was calculated by

$$ARE_j(\%) = \frac{1}{5}\sum_{i=1}^{5}\frac{\sqrt{MSE_{ij}}}{\bar{t}_{ij}} \times 100 \tag{85}$$

where MSE_{ij} is the mean squared error of jth output ith test set, and $\bar{t}_{ij}$ is the mean value of jth target in ith test set.

5.3.3. Structure of the ANNs and SNOF

In this study, the structure of the DANNs and SNOF consisted of one input layer, one hidden layer, and one output layer. The activation function in the hidden layer was chosen as a hyperbolic tangent, and that in the output layer was a linear function. The bias terms in the DANNs and SNOF were all removed. For the one hidden neuron case of the NSSM, (17) was simplified to

$$\begin{bmatrix} x_{k+1} \\ q_{x,k} \\ y_k \end{bmatrix} = \begin{bmatrix} 0 & W_{AB} & 0 \\ V_A & 0 & V_B \\ W_{CD} & 0 & 0 \end{bmatrix} \begin{bmatrix} x_k \\ p_{x,k} \\ u_k \end{bmatrix}. \tag{86}$$

In the case of the SNOF, D_{qp} and D_{yp} were set to zero to simplify the model identification algorithm.

5.3.4. Performance comparison

The general SNOF, NSSM, and GIOM are compared for 4-input 1-output modelling in terms of the ARE in Table 5 and transient responses in Figures 1–5 for the test datasets. The SNOF has the best performance for every output when the models have the same number of hidden neurons, in some cases by nearly an order of magnitude lower ARE. For the NSSM, 3 hidden neurons were required to fit the methanol concentration, and 2 hidden neurons for the glycerol and product concentrations. GIOM also needed 2 hidden neurons to fit the glycerol, methanol, and product concentrations. The SNOF is observed to be the most compact model. A SNOF with a single neuron has low prediction error for the prediction of all of the variables except for methanol concentration, in which two neurons was needed. The one-neuron DANNs had large errors for all of the concentrations, in several cases by about a factor of five.

6. Conclusions

Dynamic artificial neural networks are widely reported in industrial applications without having any theoretical guarantees of stability or performance. Academic studies have derived sufficient conditions to provide theoretical conditions for proving stability, but the results tend to be conservative for large numbers of neurons.

This article analyzes the matrix structures and degrees of freedom to assess the relative ability of the standard normal operator form and three popular dynamic artificial neural network models to fit the input-output behavior of nonlinear dynamical systems. A series of examples were used to illustrate the construction of the real matrix that specifies a SNOF, and to show that the matrix constructions for the DANNs have fewer degrees of freedom than the SNOF. For a multistage bioreactor, the SNOF is observed to be have much lower average relative error than the DANNs, when the models have the same number of neurons. Typically the SNOF resulted in accurate predictions even when having only a single neuron. The results imply that SNOF is a promising neural network model structure for the identification and control of nonlinear dynamical systems.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

CRediT authorship contribution statement

Pil Rip Jeon: Conceptualization, Data curation, Formal analysis, Funding acquisition, Investigation, Methodology, Software, Validation, Visualization, Resources, Writing – original draft, Writing – review & editing. **Moo Sun Hong:** Formal analysis, Investigation, Methodology, Software, Validation, Writing – original draft, Writing – review & editing. **Richard D. Braatz:** Conceptualization, Formal analysis, Funding acquisition, Investigation, Methodology, Project administration, Resources, Supervision, Writing – original draft, Writing – review & editing.

Acknowledgements

This work was supported the U.S. Food and Drug Administration (U01FD006483), the Basic Science Research Program of the National Research Foundation of Korea (NRF) funded by the Ministry of Education (2020R1A6A3A03039945), and the MOTIE (Ministry of Trade, Industry, and Energy) in Korea, under the Fostering Global Talents for Innovative Growth Program (P0008747) supervised by the Korea Institute for Advancement of Technology (KIAT). Any opinions, findings, conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the financial sponsors.

Appendix A

A1. Lyapunov function

Lyapunov analysis is widely used for assessing whether a nonlinear dynamical system is stable. Lyapunov's method for discrete-time systems is "A system of the form $\hat{x}_{k+1} = f(\hat{x}_k)$ is globally asymptotically stable at the origin if there exists a function V, referred to as the *Lyapunov function*, such that (1) $V(\hat{x}_{eq}) = 0$ and $V(\hat{x}_k) > 0$ for all $\hat{x}_k \neq 0$, (2) $\triangle V(\hat{x}_k) = V(\hat{x}_{k+1}) - V(\hat{x}_k) < 0$ for all $\hat{x}_k \neq 0$, and (3) $V(\hat{x}_k) \to \infty$ as $||\hat{x}_k|| \to \infty$ (Brogan, 1991)."

The application of Lyapunov analysis to Lure systems employs the Lyapunov function:

$$V(x_k) = \bar{x}_k^\top P \bar{x}_k + 2\sum_{n=1}^{n_q} \lambda_i \int_0^{q_{k,i}} \phi_i(\sigma)\mathrm{d}\sigma, \tag{87}$$

or some variation thereof, where $\bar{x}_k$ is the extended state vector

$$\bar{x}_k \triangleq \begin{bmatrix} x_k \\ p_k \\ q_k \end{bmatrix}, \tag{88}$$

and P is a positive definite matrix and λ_i are non-negative real numbers that serve as unknowns to be determined in the Lyapunov analysis. The Lyapunov function satisfies condition (1) by construction. Both p_k and q_k are functions of the state variable vector x_k, and the above Lyapunov function is radially unbounded and positive for all nonzero $x_k \in \mathbb{R}^n$, which satisfies condition (2). As shown in the second term on the right-hand side of (87), the Lyapunov equation is a linear function of the nonlinearities. When deriving an algebraic condition for ensuring that condition (3) holds, the latter terms in (87) result in a nonconvex set of inequalities, which are then convexified by applying the *S-procedure* (Boyd et al., 1994). The *S-procedure* introduces conservatism for each term in the summation in (87), which increases as the number of nonlinear terms increases (Kim, 2009).

References

Albertini, F., Sontag, E.D., Maillot, V., 1993. Uniqueness of weights for neural networks. In: Artificial Neural Networks with Applications in Speech and Vision. Chapman and Hall, pp. 115–125.
Billings, S.A., Jamaluddin, H.B., Chen, S., 1992. Properties of neural networks with applications to modelling non-linear dynamical systems. Int. J. Control 55 (1), 193–224. doi:10.1080/00207179208934232
Boyd, S., El Ghaoui, L., Feron, E., Balakrishnan, V., 1994. Linear matrix inequalities in system and control theory. SIAM, New York.
Brogan, W.L., 1991. Modern control theory. Upper Saddle River:Prentice-Hall, NJ.
Chen, C.-T., 1998. Linear system theory and design. Oxford University Press, Oxford, United Kingdom.
Cybenko, G., 1989. Approximation by superpositions of a sigmoidal function. Math. Control Signal. Systems 2 (4), 303–314. doi:10.1007/BF02551274
Desoer, C.A., Vidyasagar, M., 1975. Feedback systems: Input-Output properties. Academic Press, New York.
Fang, Y., Kincaid, T.G., 1996. Stability analysis of dynamical neural networks. IEEE Trans. Neural Netw. 7 (4), 996–1006. doi:10.1109/72.508941
FDA, 1999. Guidance for industry ANDAs: Impurities in drug substances.
Forti, M., Tesi, A., 1995. New conditions for global stability of neural networks with application to linear and quadratic-programming problems. IEEE Trans. Circuits Syst. I. Fundam. Theory Appl. 42 (7), 354–366. doi:10.1109/81.401145
Gantmacher, F., Yakubovich, V., 1966. Absolute stability of nonlinear control systems. In: Proc. II All-Union Conf. Theoret. Applied. Mech., pp. 30–63.
Hong, M.S., Velez-Suberbie, M.L., Maloney, A.J., Biedermann, A., Love, K.R., Love, J.C., Mukhopadhyay, T.K., Braatz, R.D., 2021. Macroscopic modeling of bioreactors for recombinant protein producing pichia pastoris in defined medium. Biotechnol. Bioeng. 118 (1), 1199–1212. doi:10.1002/bit.27643
Jin, L., Gupta, M.M., 1996. Globally asymptotical stability of discrete-time analog neural networks. IEEE Trans. Neural Netw. 7 (4), 1024–1031. doi:10.1109/72.508944
Khalil, H.K., Grizzle, J.W., 2002. Nonlinear systems. Prentice Hall, Upper Saddle River, New Jersey.
Kim, K.K., 2009. Robust control for systems with sector-bounded, slope-restricted, and odd monotonic nonlinearities using linear matrix inequalities. University of Illinois at Urbana-Champaign.
Kim, K.K.K., Ríos Patrón, E., Braatz, R.D., 2011. Universal approximation with error bounds for dynamic artificial neural network models: A tutorial and some new results. In: Proceedings of the IEEE International Symposium on Computer-Aided Control System Design, pp. 834–839. doi:10.1109/CACSD.2011.6044542
Kim, K.-K.K., Ríos Patrón, E., Braatz, R.D., 2018. Standard representation and unified stability analysis for dynamic artificial neural network models. Neural Netw. 98, 251–262. doi:10.1016/j.neunet.2017.11.014
La Salle, J., Lefschetz, S., 1961. Stability by Liapunov's direct method: With applications. Academic Press, New York.
Levin, A.U., Narendra, K.S., 1995. Recursive identification using feedforward neural networks. Int. J. Control 61 (3), 533–547. doi:10.1080/00207179508921916
Liberzon, M., 2006. Essays on the absolute stability theory. Autom. Remote Control 67 (10), 1610–1644. doi:10.1134/S0005117906100043
Lur'e, A.I., Postnikov, V.N., 1944. On the theory of stability of control systems. Appl. Math. Mech. 8 (3), 246–248.
Lyapunov, A.M., 1892. The general problem of the stability of motion. Moscow University, Moscow. p. 62
Monod, J., 1949. The growth of bacterial cultures. Annu. Rev. Microbiol. 3, 371–394. doi:10.1146/annurev.mi.03.100149.002103
Narendra, K.S., Parthasarathy, K., 1990. Identification and control of dynamical systems using neural networks. IEEE Trans. Neural Netw. 1 (1), 4–27. doi:10.1109/72.80202
Peri, D., Tinti, F., 2012. A multistart gradient-based algorithm with surrogate model for global optimization. Commun. Appl. Ind. Math. 3 (1). doi:10.1685/journal.caim.393
Popov, V., 1961. On absolute stability of non-linear automatic control systems. Avtomat. i Telemekh. 22 (8), 961–979.
Popov, V., 1964. Hyperstability and optimality of automatic systems with several control functions. Rev. Roum. Sci. Tech., Ser. Electrotech. Energ 9 (4), 629–690.
Qin, S.J., Badgwell, T.A., 2003. A survey of industrial model predictive control technology. Control Eng. Pract. 11 (7), 733–764. doi:10.1016/S0967-0661(02)00186-7
Ren, H., Yuan, J., Bellgardt, K.-H., 2003. Macrokinetic model for methylotrophic pichia pastoris based on stoichiometric balance. J. Biotechnol. 106 (1), 53–58. doi:10.1016/j.jbiotec.2003.08.003
Suykens, J.A.K., Moor, B.D., Vandewalle, J., 1995. Nonlinear system identification using neural state space models, applicable to robust control design. Int. J. Control 62 (1), 129–152. doi:10.1080/00207179508921536
Suykens, J.A.K., Vandewalle, J., Moor, B.D., 1996. Artificial neural networks for modeling and control of nonlinear systems. Kluwer Academic Publishers, Dordrecht, Netherlands.
Vidyasagar, M., 2002. Nonlinear systems analysis. Upper Saddle River:Prentice-Hall, NJ.
Volgis, C., Lagaris, I., 2006. A global optimization approach to neural network training. Neural, Parallel and Sci. Comput 14 (2), 231–240.
Yakubovich, V.A., 1998. A quadratic criterion for absolute stability. Dokl. Ross. Akad. Nauk 361 (5), 608–611.