

Research Article

Neural Network Inverse Model Control Strategy: Discrete-Time Stability Analysis for Relative Order Two Systems

M. A. Hussain,^{1,2} Jarinah Mohd Ali,¹ and M. J. H. Khan¹

¹ Department of Chemical Engineering, Faculty of Engineering, University of Malaya, 50603 Kuala Lumpur, Malaysia

² UM Power Energy Dedicated Advanced Center (UMPEDAC), University of Malaya, 59990 Kuala Lumpur, Malaysia

Correspondence should be addressed to M. A. Hussain; mohd_azlan@um.edu.my

Received 29 January 2014; Revised 4 April 2014; Accepted 4 April 2014; Published 12 May 2014

Academic Editor: Ju H. Park

Copyright © 2014 M. A. Hussain et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper discusses the discrete-time stability analysis of a neural network inverse model control strategy for a relative order two nonlinear system. The analysis is done by representing the closed loop system in state space format and then analyzing the time derivative of the state trajectory using Lyapunov's direct method. The analysis shows that the tracking output error of the states is confined to a ball in the neighborhood of the equilibrium point where the size of the ball is partly dependent on the accuracy of the neural network model acting as the controller. Simulation studies on the two-tank-in-series system were done to complement the stability analysis and to demonstrate some salient results of the study.

1. Introduction

The utilization of the inverse model of a plant or simply called the inverse model as controlling nonlinear systems has become increasingly useful recently [1]. Since neural networks have the well-known ability to model any system arbitrarily accurately, their use as the inverse model in this control strategy is highly appropriate [2–6]. However many of these nonlinear-based inverse control strategies including the neural network inverse model control strategy are normally formulated in input-output form. These input-output approaches have some important input-output stability and feedback properties but do not lead to an analytical controller formulation and global stability analysis, given the abstract nature the nonlinear operators applied (in contrast to linear systems where output feedback controllers can be derived in the form of transfer functions). Synthesis and analysis of nonlinear dynamic output feedback controllers including closed loop stability analysis normally rely on the state space realizations of the controllers within these nonlinear systems. Furthermore, in nonlinear systems,

much more useful process information is hidden in the state space description of the process.

Stability theory plays a central role in the design and application of various control algorithms and one popular method to study stability especially for nonlinear system is the Lyapunov stability theory. This theory states that an equilibrium point is asymptotically stable if all solutions starting at nearby points not only stay nearby but also tend to the equilibrium points as time approaches infinity. This is normally done by defining the Lyapunov function and checking on the value of its derivative with time. The details of this method can be found in [7, 8]. This theory has been applied for many other control algorithms in recent years [9–12] but limited work has been found in the literature concerning the stability analysis of inverse model neural network control strategies.

Hernández and Arkun [13] studied the local stability of these neural network inverse model control strategies by analyzing the Jacobian of the state space matrix. Etxebarria [14] proved the closed loop stability of their neural network based adaptive system through studying the convergence of

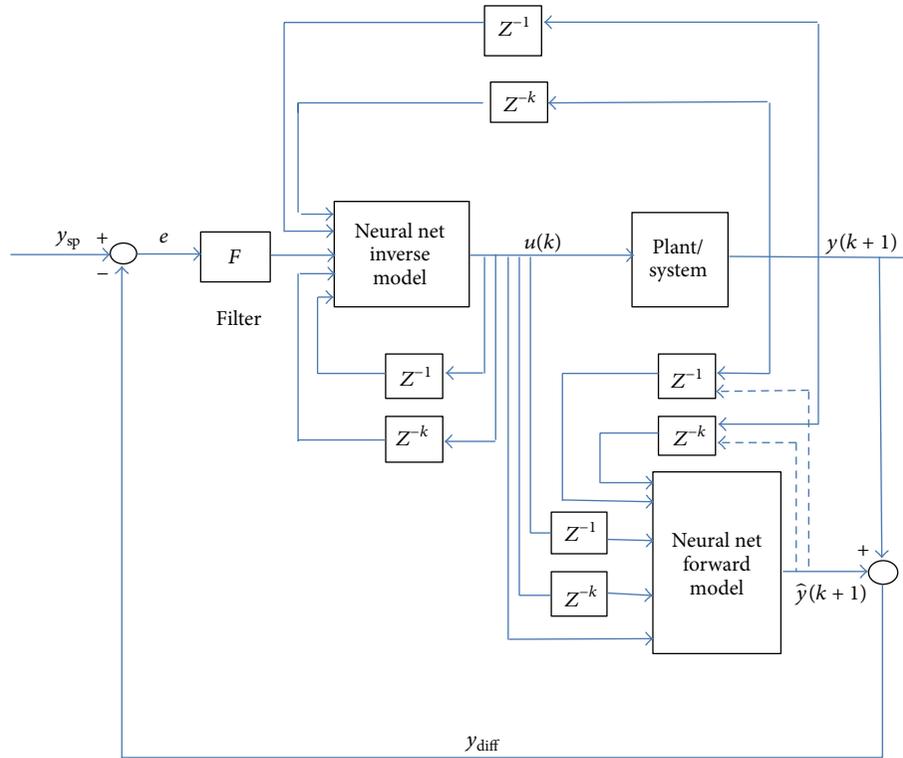


FIGURE 1: Neural network in IMC strategy.

the error bounds in tracking. Nikravesh et al. [15] also studied the stability analysis of a dynamic neural network based control system in an internal model control (IMC) framework by also analyzing the eigenvalues of the Jacobian of the state space matrix representations, which is then used to determine the optimal neural network structure. Recently, Piroddi utilized a Lyapunov based method to analyze the properties of a hybrid neural network controller [16] and also Wu et al. used a Lyapunov method to study the stability of an adaptive neural-based controller for control of an inverted pendulum [17].

The difficulty in analyzing the stability of the neural network control strategies is that it is difficult to formulate the neural network controller in the format where the Lyapunov based stability analyses can be applied directly since neural networks are normally represented by an input-output structure only. Hence, our attempt here differs from the rest in that it involves the representation of the neural network inverse model control system in state space format and the direct usage of the Lyapunov second method to study the closed loop stability of the system, which is one novel contribution of this work.

The structure of the paper is as follows. Initially we discuss the inverse model control strategy involving neural networks. Next we discuss the formulation of the closed loop state space representation for the inverse model control strategy controlling typical relative order two systems. This is followed by discussing the stability analysis of the closed loop systems through Lyapunov's method, after which the analysis is complemented through some simulation case studies on a two-tank-in-series system.

2. Neural Network Inverse Model Control Strategy

A simple form in utilizing neural networks for control in the inverse model control strategy is the direct inverse control strategy. In this case the neural network inverse model, acting as the controller, has to learn to supply at its output the appropriate control parameters to achieve the desired target input. The network inverse model is then utilized in the control strategy by simply cascading it with the controlled system or plant. In this control scheme the desired set point is fed to the network together with the past plant inputs and outputs to predict the desired current input [18]. These methods rely heavily on the accuracy of the inverse model and lack robustness, which can be attributed mainly to the absence of the feedback.

A much more robust and stable strategy is that of the non-linear internal model control technique, which is basically an extension of the linear IMC method [19]. In this method both the neural network forward and inverse models (see Figure 1) are used directly as elements within the feedback loop. The neural network inverse model can be obtained by directly pretraining the neural network to identify the inverse of the process model or by numerically inverting the neural network forward model at each interval by Newton's method. However these numerical techniques are computationally intensive and time consuming and are highly sensitive to the initial estimates. The use of the pretrained neural networks model gives faster implementation, and we will resort to this method in our simulation studies later. Details of these forward and

inverse models as well as their method of identification can be found elsewhere [18, 20–23]. The IMC approach is similar in structure to the direct inverse approach above except for two additions. First is the addition of the forward model placed in parallel with the plant, to cater for plant or model mismatches, and second is that the error between the plant output and neural net forward model is subtracted from the set point before being fed into the inverse model. The other input data to the inverse model is similar to the direct inverse method. The forward model is fed with the input of the plant (i.e., output of inverse model) as well as the past inputs and past outputs of the plant in our study. The forward model can however also be fed with its own past outputs instead of the plant outputs to form a recurrent network, especially in dealing with cases of noisy plant output data (as seen from the dotted line in Figure 1). A tuning filter F is introduced prior to the controller in this IMC approach to incorporate robustness in the feedback system (especially where it is difficult to get exact inverse models) and also to project the error signal into the appropriate input space of the controller.

3. Closed Loop State Space Representation of Inverse Model Control Strategy

3.1. System Representation: Relative Order Two Systems. In this case the system to be controlled is in the following form:

$$\begin{aligned} x(k+1) &= \Phi[x(k), u(k)], \\ y(k) &= h[x(k)], \end{aligned} \quad (1)$$

where in this case

$$\begin{aligned} x_n(k) &= \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix}, \\ \Phi[x(k), u(k)] &= \begin{bmatrix} f_1(x_1(k), x_2(k)) + u(k) \\ f_2(x_1(k), x_2(k)) \end{bmatrix}, \\ y &= x_2(k), \end{aligned} \quad (2)$$

where x denotes the vector of state variables, u denotes the manipulated input, and y represents the output. It is assumed that $x \in X \subset \mathfrak{R}^n$ and $u \in U \subset \mathfrak{R}$, where X and U are open-connected sets that contain the origin (i.e., the nominal equilibrium point). $\Phi(x, u)$ is an analytic vector function on $X \times U$ and $h(x)$ is an analytic scalar function on X .

This system in discrete-time constitutes relative order two systems since it satisfies the criteria of $\partial_y(k+1)/\partial_u(k) \equiv 0$ and $\partial_y(k+2)/\partial_u(k) \neq 0$ [24].

3.2. State Space Representation: Neural Network Inverse Model.

Next we can represent the neural network inverse model as a state space realization in terms of vector $z(k)$, which comprises all the states and inputs utilized in the network, that is, past and present states as well as the past inputs (refer to Figure 1), as follows:

$$\begin{aligned} z_1^y(k) &= x_1(k - n_y), \\ z_2^y(k) &= y(k - n_y), \end{aligned}$$

$$\begin{aligned} z_3^y(k) &= x_1(k - n_{y+1}), \\ z_4^y(k) &= y(k - n_{y+1}), \\ &\vdots \\ z_{2n_y+1}^y(k) &= x_1(k), \\ z_{2n_y+2}^y(k) &= y(k), \\ z_{2n_y+3}^y(k) &= y(k+1), \\ z_{2n_y+4}^y(k) &= e_f(k), \\ z_1^u(k) &= u(k - n_u), \\ &\vdots \\ z_{n_u}^u(k) &= u(k - 1), \end{aligned} \quad (3)$$

where $e_f(k) = y_{sp} - (y(k+1) - \hat{y}(k+1))$, y_{sp} is the set point, $y(k+1)$ is the plant output, and $\hat{y}(k+1)$ is the model output. It is also assumed in this case that $|y(k+1) - \hat{y}(k+1)| \leq \phi|y(k) - \hat{y}(k)|$, where $\phi \in (0, 1)$. The closed loop enlarged state space representation of the system can then be obtained by incorporating the closed loop equation in this neural network state space representation. This closed loop equation in terms of the output, $y(k)$, can be obtained analytically as will be shown in the next section.

3.3. Closed Loop Formulation of Inverse Model Based Control Strategy Perfect Model Case. In the inverse model strategy such as the IMC method, the output from the neural network model can be written in the equivalent prediction form as

$$\begin{aligned} y(k + \Theta + 1) &= f(y(k + \Theta), \dots, y(k + \Theta - n_y + 1), \\ &u(k), \dots, u(k - n_u + 1)), \end{aligned} \quad (4)$$

where Θ is the assumed time delay between input and output and the systems and has relative order greater than one [25] and f is a nonlinear function of the present and past states as well as the present and past inputs of the system.

The output of the inverse neural network model, $u(k)$, acting as the controller can be represented by the following equation:

$$\begin{aligned} u(k) &= \hat{g}(y(k + \Theta + 1), y(k + \Theta), \dots, y(k + \Theta - n_y + 1), \\ &u(k - 1), \dots, u(k - n_u + 1)), \end{aligned} \quad (5)$$

where \hat{g} is a nonlinear function of the present and past states, the required output, and the past inputs of the systems. This value of $u(k)$ can be obtained by solving the equation above analytically for the current input or directly from the pretrained neural network inverse model. Suppose that the following response is required:

$$y(k + \Theta + 1) = v(k), \quad (6)$$

where z_d is defined as the set point or equilibrium values of the state space variables at time $(k + 1)$,

Where,

$$\bar{A} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & \dots & \dots & 0 \\ 0 & 0 & 0 & 1 & \vdots \\ 0 & 0 & 0 & 0 & 1 & \vdots \\ 0 & 0 & 0 & 0 & 0 & 1 & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ \vdots & 1 & \vdots & \vdots & \vdots \\ \vdots & 1 & \vdots & \vdots \\ 0 & 0 & 0 & \dots & \dots & \dots & \dots & \dots & A_c & \alpha & 0 & 0 \\ 0 & 0 & 0 & \dots & \dots & \dots & \dots & \dots & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & \dots & \dots & \dots & \dots & \dots & 0 & \alpha & 0 & 0 \\ 0 & 0 & 0 & \dots & \dots & \dots & \dots & \dots & 0 & 0 & \phi & 0 \\ & & & & & & & & & 0 & 1 & 0 & \dots & 0 \\ & & & & & & & & & 0 & 0 & 1 & \dots & \vdots \\ & & & & & & & & & \vdots & \vdots & \vdots & \vdots & \vdots \\ & & & & & & & & & \vdots & \vdots & \vdots & \vdots & \vdots \\ & & & & & & & & & \vdots & \vdots & \vdots & \vdots & 1 \\ & & & & & & & & & 0 & \dots & \dots & \dots & 0 \end{bmatrix}, \tag{15}$$

$$\bar{D} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ (1 - \alpha) \Delta y_{sp}(k) \\ 0 \\ \Delta C_c \\ 0 \\ 0 \\ 0 \\ \vdots \\ \vdots \\ \vdots \\ \Delta u(k) \end{bmatrix}. \tag{16}$$

Also,

$$\Delta C_c = (1 - \alpha) \Delta y_{sp}(k), \tag{17a}$$

$$A_c = -(1 - \alpha). \tag{17b}$$

Note that α is the tuning parameter of the associated filter. The value of α is set in the range of 0 to 1 [17], which also ensures the stability of the matrix A (a condition required for the stability analysis later). The actual value of α however affects the shape and speed of responses, which will be demonstrated in the simulation studies later. The

closed loop representation of (14) will then be analyzed for its stability properties by the use of Lyapunov's method in the next section.

4. Closed Loop Stability Analysis of Inverse Model Control Strategy

Assumptions. (1) The desired state sequence $z_d(k)$ is uniformly bounded in that

$$|z_d^i(k)| \leq m_i < \infty, \quad \forall i, \forall k \geq 0. \quad (18)$$

(2) The error between the model and plant decreases with the time according to

$$|y(k+1) - \hat{y}(k+1)| \leq \phi |y(k) - \hat{y}(k)| \quad (19)$$

where $\phi \in (0, 1)$.

(3) The approximation of the neural network inverse model (acting as the controller), which is given by $|u(k) - u_d(k)| \leq k_u$, is very small such that $(k_u)^2$ can be neglected.

We can then state the theorem below.

Theorem 1. *For specific set points ($\Delta y_{sp}(k) = 0$), the error of the nonlinear systems (1) as defined by $\Delta z(k)$, under the neural network inverse model control law, is confined to a neighborhood of the origin in that $\|\Delta z\| \leq \epsilon_{k2}$, for all $k \geq 0$, for some finite ϵ_{k2} .*

Proof. First consider the Lyapunov function associated with $\Delta z(k)$ as

$$V(\Delta z(k)) = \Delta z^T(k) \bar{P} \Delta z(k), \quad (20)$$

where \bar{P} is a positive, definite, symmetric matrix solution of the following discrete-time matrix Lyapunov equation:

$$\bar{A}^T \bar{P} \bar{A} - \bar{P} = -\bar{Q}. \quad (21)$$

Here \bar{A} is a discrete-time stable matrix of (15), and \bar{Q} is a symmetric matrix, where $\bar{Q} > 0$.

Let

$$\Delta_k V(\Delta z(k)) = V(\Delta z(k+1)) - V(\Delta z(k)). \quad (22)$$

From (14), we get

$$\begin{aligned} \Delta_k V(\Delta z(k)) &= (\bar{A} \Delta z(k) + \bar{D})^T \bar{P} (\bar{A} \Delta z(k) + \bar{D}) \\ &\quad - (\Delta z(k))^T \bar{P} (\Delta z(k)), \end{aligned} \quad (23)$$

where \bar{A} and \bar{D} are as defined previously.

Applying Lyapunov's discrete-time equation, we simplify (23) to

$$\begin{aligned} \Delta_k V(\Delta z(k)) &= 2(\Delta z(k))^T \bar{A}^T \bar{P} \bar{D} + \bar{D}^T \bar{P} \bar{D} \\ &\quad - (\Delta z(k))^T \bar{Q} (\Delta z(k)). \end{aligned} \quad (24)$$

Hence, incorporating (14), with the above simplification, into (24), we obtain

$$\begin{aligned} \Delta_k V(\Delta z(k)) &= 2(\Delta z(k))^T \bar{A}^T \bar{P} ([\Delta C_c] e_i \\ &\quad + [(1-\alpha) \Delta y_{sp}(k)] e_k \\ &\quad + [\Delta u(k)] e_n) \\ &\quad + ([\Delta C_c] e_i + [(1-\alpha) \Delta y_{sp}(k)] e_k \\ &\quad + [\Delta u(k)] e_n)^T \\ &\quad \times \bar{P} ([\Delta C_c] e_i + (1-\alpha) \Delta y_{sp}(k) e_k \\ &\quad + [\Delta u(k)] e_n) \\ &\quad - (\Delta z(k))^T \bar{Q} (\Delta z(k)), \end{aligned} \quad (25)$$

where e_i , e_k , and e_n are the i th, k th, and n th standard basis vectors.

In the stability analysis, we are interested in the stability at the definite set and equilibrium point, y_{sp} , so that $\Delta y_{sp}(k) = 0$.

Therefore, (25) becomes

$$\begin{aligned} \Delta_k V(\Delta z(k)) &= 2(\Delta z(k))^T \bar{A}^T \bar{P} \Delta u(k) e_n \\ &\quad + (\Delta u(k))^2 e_n^T \bar{P} e_n - (\Delta z(k))^T \bar{Q} \Delta z(k). \end{aligned} \quad (26)$$

We can then get the bounds on $\Delta_k V$ by analyzing the norms of the various terms, respectively, as

$$\begin{aligned} \Delta_k V(\Delta z(k)) &\leq -\bar{Q}_{\min} \|\Delta z(k)\|^2 \\ &\quad + 2 \|\Delta z(k)\| \|\bar{A}\| \|\bar{P}\| |\Delta u(k)| \|e_n\| \\ &\quad + |\Delta u(k)|^2 \|e_n\| \|\bar{P}\| \|e_n\|, \end{aligned} \quad (27)$$

where \bar{Q}_{\min} denotes the smallest eigenvalue of \bar{Q} .

If k_u is small as discussed in the assumption, k_u^2 becomes negligible and we get

$$\Delta_k V \leq -\|\Delta z(k)\| (Q_{\min} \|\Delta z(k)\| - 2 \|A\| \|P\|), \quad (28)$$

where $\Delta_k V$ can be assured to be nonincreasing when

$$\|\Delta z(k)\| \geq \frac{2 \|A\| \|P\| |u(k)|}{Q_{\min}}. \quad (29)$$

This result shows the stability of the system, under the inverse control law, in a region of a ball which is dependent on the size $\Delta z(k)$. This size gets smaller as the value $\Delta u(k)$ gets smaller; that is, the accuracy of the neural network inverse model prediction improves and is close to the actual required value, $u_d(k)$. In fact, if the prediction error goes to zero, that is, $\Delta u(k) = 0$, asymptotic stability is achieved. \square

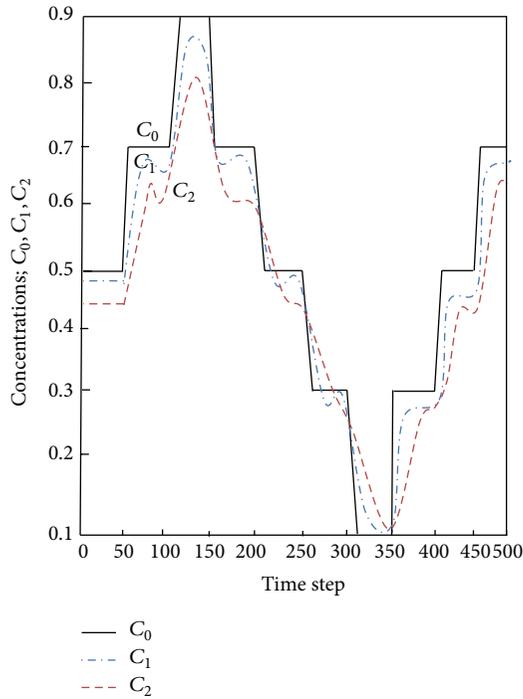


FIGURE 2: Open loop training data for neural network model—1st data set.

5. Simulation Studies

To complement the analysis performed in the previous section, simulations were performed on typical relative two nonlinear process systems using the IMC neural network based strategy. This case study involves the dynamic change in concentration of component A in a stream passing through two noninteracting thermal hold-up tanks in series, where in addition it undergoes 2nd order reactions in both tanks, respectively.

The continuous differential equations which represent the dynamics for the two tanks, respectively, are

$$\begin{aligned}
 V * \frac{dc_1}{dt} &= F * (C_0 - C_1) - k_r * C_1^2 * V, \\
 V * \frac{dc_2}{dt} &= F * (C_1 - C_2) - k_r * C_2^2 * V.
 \end{aligned}
 \tag{30}$$

In this case of inlet concentration, C_0 is the control variable [$u(k)$], C_1 is the concentration in the first tank and the concentration in the second tank, C_2 is the output variable [$y(k)$]. The values of the flow rate, F , volume of the identical tanks, V , and the rate constant, k_r , are 0.01, 0.1, and 0.01, respectively. All units are assumed to be consistent and no disturbances are present in the system.

The control of concentration in the second tank by manipulating the inlet concentration to the first tank, in discrete-time formulation, constitutes relative order two systems. In this simulation study, the pretrained inverse neural

model and the pretrained neural network forward model were incorporated in the IMC strategy, as shown in Figure 1. The models were trained using multistep data sequence with the output concentration, C_2 , in the range of 0.1 to 0.75, as shown in Figure 2. The details of the method of training these neural networks models can be found in [20–22]. The inverse model was modelled by multilayered feedforward neural network with 25 hidden nodes while the forward model has 20 hidden nodes. The hyperbolic tangent is used as the activation function for both the forward and inverse model networks. In the simulation, the one-step-ahead prediction and implementation are used for both forward and inverse models, respectively, as this is found to be adequate for this application and to be in accordance with the way of training these neural network models.

The first simulation involves controlling the output concentration at the nominal steady state value of 0.456 as well as at lower and higher step values, using the IMC strategy above. The control sampling time is chosen to be equivalent to the data sampling time of 10 sec, and each time step in the plots represents this time period of implementation. These results, as shown in Figure 3, show good asymptotic set point tracking with minimal offsets at all points except at the higher value of 0.8. This tracking behavior was predicted by the theoretical analysis for a sufficiently well-trained inverse model, acting as the controller. The higher set point value of 0.8 was however outside the training range of the inverse model and hence the model was not able to predict a good control value at this point, which results in a high offset value, also in accordance with the theoretical analysis done earlier. This problem was alleviated by retraining the inverse model with a wider output concentration range of 0.5 to 0.83 for C_2 , as shown in Figure 4, and then implementing it back in the IMC strategy as before. This time the results, as can be seen in Figure 5, show good asymptotic tracking at the higher set point value of 0.8 with minimal offsets as with the other values. These simulation results show that a well-trained neural network inverse model (which includes training the network within a sufficient and appropriate range of training data) gives good prediction of $u(k)$, that is, small value of $\Delta u(k)$, which results in smaller offsets as predicted in the stability analysis of the earlier section.

The next simulation in this case study shows the effect of the filtering action on the tracking performance. In this case, the servo performance of the closed loop systems was demonstrated for set point tracking of the concentration beginning from the steady state value of 0.456 to a lower (0.3) and higher value (0.6), using different tuning values of α , that is, 0.96 and 0.7, respectively. The results can be seen in Figures 6 and 7, respectively. Both results gave good asymptotic tracking of the set points with minimal offsets, as in the previous simulation. However the result for α at 0.96 showed sluggish response and control action, while the result for α at 0.7 showed fast control actions with excellent response. Hence, these results show that although the filtering action does not directly affect the stability of the systems (as long as it is in the range of 0 to 1), it does affect the shape and speed of tracking of the response as would be expected of a filter.

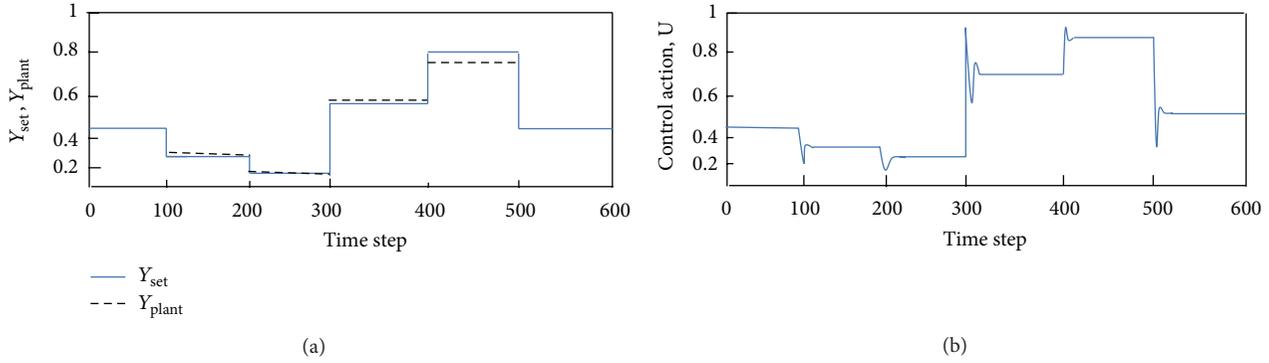


FIGURE 3: Set point tracking with IMC strategy—1st data set.

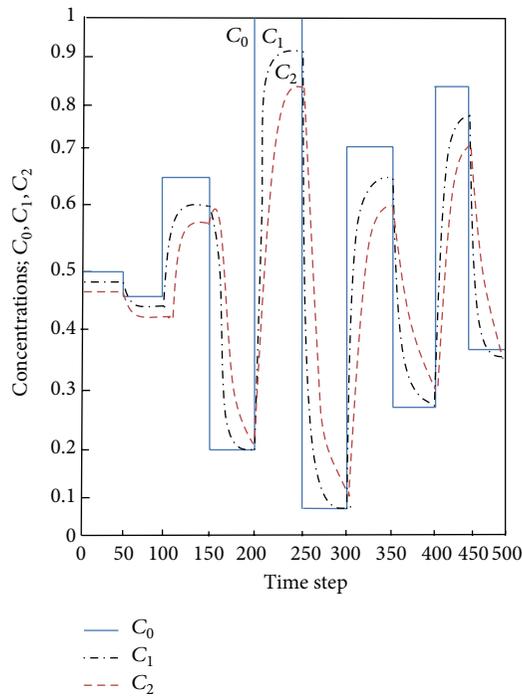


FIGURE 4: Open loop training data for neural network model 2nd data set.

The final simulation is done to validate the closed loop behavior of the IMC strategy, as derived in the theoretical analysis. This was done by plotting with time the expression on the left-hand side of the reformulated closed loop equation for relative order two systems, to see if it equals the right-hand side for this particular case study. The reformulated closed loop equation is given below:

$$\frac{(y(k+2) - \alpha y(k+1))}{(y_{sp}(k))} = (1 - \alpha). \quad (31)$$

The results, for one specific tuning constant of 0.04, can be seen in Figure 8. It gave a constant value of 0.96 (value of the

right-hand side of the equation above) with slight, expected deviations at the step changes. Hence this shows that the closed loop behavior of the neural network IMC strategy in this case study behaves according to the equation derived in the analysis earlier, which justifies its use in the closed loop state space representation.

6. Conclusions

This work is a novel, direct attempt at providing the framework and guideline for directly analyzing the stability of nonlinear controlled systems involving neural network inverse models by the use of conventional Lyapunov stability

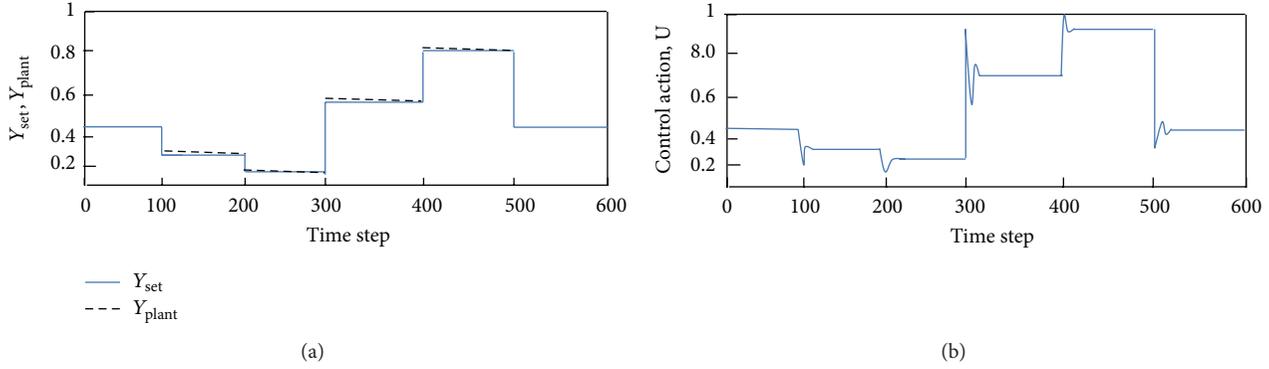


FIGURE 5: Set point tracking with IMC strategy—2nd data set.

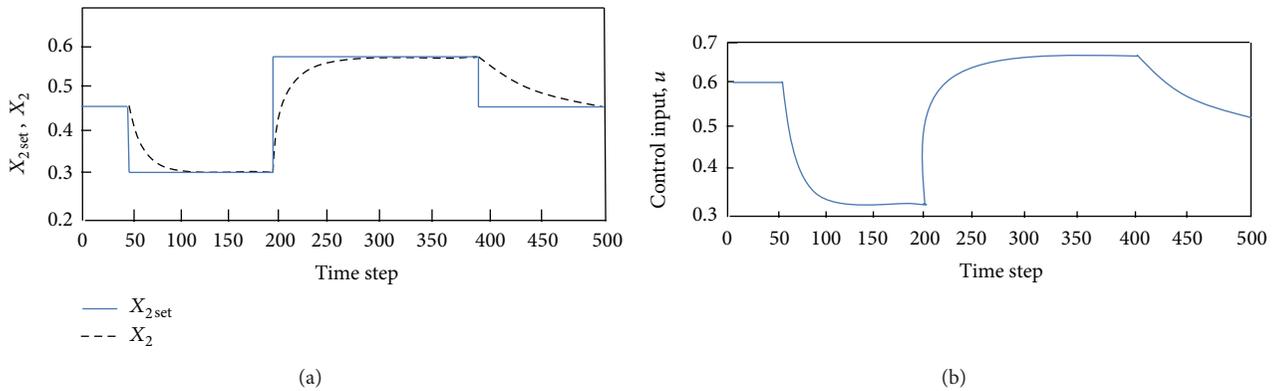


FIGURE 6: Set point tracking with tuning constant, $\alpha = 0.96$.

techniques. The results of the analysis showed bounded tracking error converging to a small ball around the equilibrium point, that is, “ball stability.” The size of the ball is shown to be dependent on the accuracy of the neural network inverse model, acting as the controller. In fact, if the prediction is perfect, it can be shown that asymptotic stability is achieved. This all makes intuitive sense, as it would be expected that the performance, tracking error, and stability of this type of control strategy would be directly dependent on the accuracy of the inverse model acting as the controller. The simulation result on a typical relative order two process system complements the analysis by showing the effect of the accuracy of the neural network controller on the system’s performance. It also demonstrates the servo performance with varying tuning parameters. The results show that when the tuning parameter was tuned properly, excellent tracking and smooth control were achieved. Finally the validation of the closed loop equation of this IMC approach was also shown in the simulation. The theoretical analysis also lends itself as the basis for further analyzing systems which utilizes both known models (such as in GMC and GLC techniques) and neural networks in a hybrid fashion within a feedforward-feedback control methodology for better treatment of model mismatches and disturbances.

Notations

\bar{A} :	Closed loop state matrix
A_c :	Constant
C_0, C_1, C_2 :	Concentration of inlet, first tank, and second tank
ΔC_c :	Constant
\bar{D} :	Closed loop state space vector
e :	Error signal
e_f :	Filtered error signal
e_i, e_k, e_n :	Standard basic vector
F :	Flow rate in tank system
k_r :	Reaction rate constant
k_u, m_i :	Finite constant
\bar{P} :	Positive definite solution of Lyapunov equation
\bar{Q} :	Symmetric matrices
r :	Relative order of controlled output, y
u :	Manipulated variable
Δu :	Deviation variable (from required value) for u
$V(k)$:	Filter output
V :	Volume of tank
$\Delta_k V$:	Derivative of V in discrete-time step, k

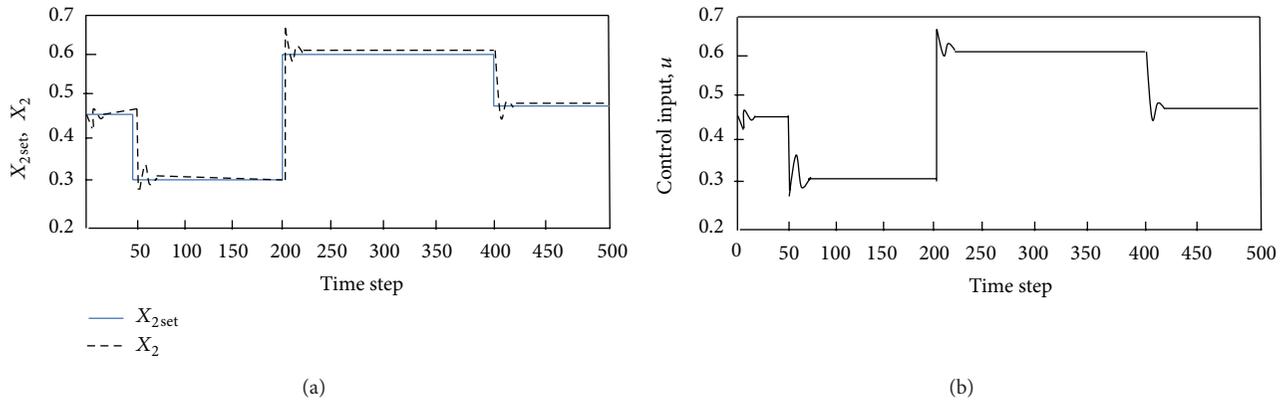


FIGURE 7: Set point tracking with tuning constant, $\alpha = 0.70$.

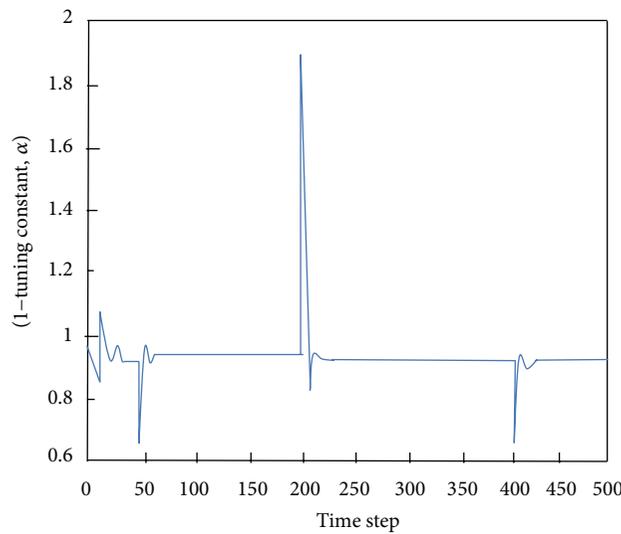


FIGURE 8: Validation of closed loop equation—plot of $(1 - \alpha)$ with time, $\alpha = 0.04$.

- x : State variable
- \bar{x} : State variable obtained from observer
- Δx : Deviation variable (from required value) for x
- y : Output variable
- \hat{y} : Output variable obtained from observer
- Δy : Deviation variable (from required value) for y
- z : State variable for neural network state space representation
- Δz : Deviation variable (from required value) for z .

Subscripts

- d Required (set) value
- sp: Set point.

Symbols

- α Tuning/filter constant
- ϕ : Finite constant.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

The authors are grateful to the University of Malaya and the Ministry of Higher Education of Malaysia (MOHE) for supporting this research project via the research Grant UM.C/HIR/MOHE/ENG/25 which made the publication of this paper possible. The authors would also like to thank the services of the research assistant Suhana Mat Idris for editing parts of the paper.

References

- [1] B. W. Bequette, "Nonlinear control of chemical processes: a review," *Industrial and Engineering Chemistry Research*, vol. 30, no. 7, pp. 1391–1413, 1991.
- [2] G. Cybenko, "Approximation by superpositions of a sigmoidal function," *Mathematics of Control, Signals, and Systems*, vol. 2, no. 4, pp. 303–314, 1989.
- [3] K. J. Hunt and D. Sbarbaro, "Studies in neural network based control," in *Neural Networks for Control and Systems*, K. Warwick, G.W. Irwin, and K.J. Hunt, Eds., IEE Control Engineering Series, pp. 94–110, London, UK, 1992.
- [4] M. A. Hussain, "Review of the applications of neural networks in chemical process control—simulation and online implementation," *Artificial Intelligence in Engineering*, vol. 13, no. 1, pp. 55–68, 1999.
- [5] I. M. Mujtaba and M. A. Hussain, "Optimal operation of dynamic processes under process-model mismatches: application to batch distillation," *Computers and Chemical Engineering*, vol. 22, no. 1, pp. S621–S624, 1998.
- [6] M. A. Hussain and L. S. Kershenbaum, "Implementation of an inverse-model-based control strategy using neural networks on a partially simulated exothermic reactor," *Chemical Engineering Research and Design*, vol. 78, no. 2, part 1, pp. 299–311, 2000.
- [7] J. J. E. Slotine and W. Li, *Applied Nonlinear Control*, Practice Hall, 1991.
- [8] H. K. Khalil, *Nonlinear Systems*, Practice Hall, 1996.
- [9] P. Vadivel, R. Sakthivel, K. Mathiyalagan, and P. Thangaraj, "Robust stabilisation of non-linear uncertain Takagi-Sugeno fuzzy systems by H_∞ control," *IET Control Theory & Applications*, vol. 6, no. 16, pp. 2556–2566, 2012.
- [10] K. Mathiyalagan and R. Sakthivel, "Robust stabilization and H_∞ control for discrete-time stochastic genetic regulatory networks with time delays," *Canadian Journal of Physics*, vol. 90, no. 10, pp. 939–953, 2012.
- [11] K. Mathiyalagan, R. Sakthivel, and S. M. Anthoni, "New robust exponential stability results for discrete-time switched fuzzy neural networks with time delays," *Computers & Mathematics with Applications*, vol. 64, no. 9, pp. 2926–2938, 2012.
- [12] S. Lakshmanan, K. Mathiyalagan, J. H. Park et al., "Delay dependent H_∞ state estimation of neural networks with mixed time-varying delays," *Neurocomputing*, vol. 129, pp. 392–400, 2014.
- [13] E. Hernández and Y. Arkun, "Study of the control-relevant properties of backpropagation neural network models of nonlinear dynamical systems," *Computers and Chemical Engineering*, vol. 16, no. 4, pp. 227–240, 1992.
- [14] V. Etxebarria, "Adaptive control of discrete systems using neural networks," *IEE Proceedings: Control Theory and Applications*, vol. 141, no. 4, pp. 209–215, 1994.
- [15] M. Nikravesh, A. E. Farrell, and T. G. Stanford, "Dynamic neural network control for non-linear systems: optimal neural network structure and stability analysis," *Chemical Engineering Journal*, vol. 68, no. 1, pp. 41–50, 1997.
- [16] L. Piroddi, "Hybrid neural control systems: some stability properties," *Journal of the Franklin Institute*, vol. 349, no. 3, pp. 826–844, 2012.
- [17] Q. Wu, N. Sepehri, and S. He, "Neural-based control and stability analysis of a class of nonlinear systems: base-excited inverted pendulums," *Journal of Intelligent and Fuzzy Systems*, vol. 12, no. 2, pp. 119–131, 2002.
- [18] Y.-H. Pao, S. M. Phillips, and D. J. Sobajic, "Neural-net computing and the intelligent control of systems," *International Journal of Control*, vol. 56, no. 2, pp. 263–289, 1992.
- [19] C. G. Economou, M. Morari, and B. O. Palsson, "Internal model control—extension to nonlinear systems," *Industrial & Engineering Chemistry Process Design and Development*, vol. 25, no. 2, pp. 403–411, 1986.
- [20] K. J. Hunt and D. Sbarbaro, "Neural networks for nonlinear internal model control," *IEE Proceedings D: Control Theory and Applications*, vol. 138, no. 5, pp. 431–438, 1991.
- [21] S. Billings and S. Chen, "Neural networks and systems identification," in *Neural Networks for Control and Systems*, IEE Control Engineering Series no. 46, pp. 181–202, 1992.
- [22] G. A. Montague, A. J. Morris, and M. J. Willis, "Artificial neural networks: methodologies and applications in process control," in *Neural Networks for Control and Systems*, IEE Control Engineering Series no. 46, pp. 123–147, 1992.
- [23] B. S. Dayal, P. A. Taylor, and J. F. Macgregor, "Design of experiments, training and implementation of nonlinear controllers based on neural networks," *Canadian Journal of Chemical Engineering*, vol. 72, no. 6, pp. 1066–1079, 1994.
- [24] H. Nijmeijer and J. V. D. Shaft, "Discrete-time nonlinear control systems," in *Nonlinear Dynamical Control Systems*, pp. 437–462, Springer, 1990.
- [25] E. P. Nahas, M. A. Henson, and D. E. Seborg, "Nonlinear internal model control strategy for neural network models," *Computers and Chemical Engineering*, vol. 16, no. 20, pp. 1039–1057, 1992.
- [26] M. Soroush and C. Kravaris, "Discrete-time nonlinear controller synthesis by input/output linearization," *AIChE Journal*, vol. 38, no. 12, pp. 1923–1945, 1992.
- [27] P. Daoutidis and C. Kravaris, "Dynamic output feedback control of minimum-phase nonlinear processes," *Chemical Engineering Science*, vol. 47, no. 4, pp. 837–849, 1992.