



Identification of Systems Having Unstable Dynamics and Time Delays Using Delayed Recurrent Neural Networks

Sudeep Sharma¹ · S. V. S. Prasad² · T. S. Arulananth²

Received: 25 September 2022 / Accepted: 17 August 2023 / Published online: 30 October 2023
 © King Fahd University of Petroleum & Minerals 2023

Abstract

This article addresses the challenging problem of identifying unstable system dynamics with time delay. In the proposed novel scheme, a recurrent neural network (RNN) with parallel delayed architecture in closed-loop has been employed, which is referred to as closed-loop-delayed-RNN (CLDRNN). The systematic mathematical formulation is done in easy to follow steps to calculate all the parameters of unstable delayed process models. Interestingly, in proposed algorithm, all model parameters are directly estimated in terms of the optimized CLDRNN weights only, without using any prior knowledge about the unknown process dynamics. The Lyapunov theory is incorporated to get efficient learning, and an accurate condition is derived to achieve guaranteed global convergence of the proposed algorithm. Various identification experiments are conducted on benchmark unstable process examples to show the efficacy of the proposed approach.

Keywords System identification · Unstable system · Time delay · Recurrent neural networks · Parameters estimation · Optimization and convergence

List of Symbols

$f(\cdot)$	Activation function	g	Gradient vector
C	Controller model	I	Identity matrix
F	Cost function	$r_1(n)$	Input at controller's output
F^*	Cost function goal	$u(n)$	Input signal
K_d	Derivative gain	$\hat{\theta}_l$	Integer delay
$\varepsilon(n)$	Difference error	K_i	Integral gain
\mathcal{E}	Difference error vector	J	Jacobian matrix
n	Discrete time	$\mathcal{L}(\cdot)$	Laplace transform
$d(n)$	Disturbance signal	V	Lyapunov function
$e(n)$	Error signal	ΔV	Lyapunov function deviation
\hat{G}	Estimated model	k_{\max}	Maximum epochs
$\hat{y}(n)$	Estimated output signal	g_{\min}	Minimum gradient
$\hat{\theta}_f$	Fractional delay	m	Noise mean
β	Filter coefficient	P_{noise}	Noise power
\hat{K}	Gain	P_{signal}	Noise power
		σ^2	Noise variance
		N	Number of training samples
		$y(n)$	Output signal
		∂	Partial derivative operator
		G_p	Process model
		K_p	Proportional gain
		$r(n)$	Reference input
		μ	Scalar coefficient
		μ_{\max}	Scalar's maximum limit
		μ_f	Scalar multiplier

✉ Sudeep Sharma
 sudeep.sharma@iiitsurat.ac.in

S. V. S. Prasad
 drsvs.prasad@mlrinstitutions.ac.in

T. S. Arulananth
 arulananth.ts@mlrinstitutions.ac.in

¹ Indian Institute of Information Technology, Surat, India

² MLR Institute of Technology, Hyderabad, India



Σ	Summation operator
\hat{t}	Time constant
$\hat{\theta}$	Time delay
k	Training epoch
T	Transpose
w	Weight
ΔW	Weight deviation vector
W	Weight vector
M	Weight vector length
\mathcal{Z}	[.] Z-transform

1 Introduction

The data-driven system identification has been extensively used to model systems having complex and nonlinear dynamics [1, 2]. The time delay occurs naturally in almost all industrial process models, where the material or information takes some time to travel between source and destination [3]. The problem with time delay estimation is that it produces multiple minima in the cost function, which makes global minima hard to reach [4].

In general, many industrial processes are preferred to be identified as lower order models with time delays [5–10]. There are various model-based techniques available which uses identified lower-order (mostly first- or second-order) time delay models to design and tune regulators [11–19]. Therefore, an accurate model estimation plays a key role in the success of these model-based techniques.

The literature studies suggest that the identification of stable delayed dynamics is relatively easy and extensively available in various papers (see [20] and references therein). The advantage associated with the stable delayed process is that the identification can be performed in open-loop with less concern on the predictor's stability. On the other hand, the identification of unstable delayed dynamics is a challenging task, which should be carried out in closed-loop to keep the output bounded. In general, identification with a controller in the loop is referred to as closed-loop identification [21]. More specifically, a regulator is needed to stabilize unstable dynamic systems, which also helps to ensure the safety concerns and in generating stable predictors [22]. Moreover, most of the methods that are useful for stable delayed process identification may not be suitable for unstable counterparts and hence the availability of such methods is limited.

Although neural networks have been used in various modeling and control-related tasks [23, 24] and, also, in handling time delays effectively as well [25–30], however, the existing contributions that uses neural networks for estimating system models are nonparametric due to uncertainty in selecting layers, number of neurons and type of activation functions [31, 32]. Another limitation associated with the neural networks is that it may produce overfitted models [33]. An identified

model is considered to be overfitted if it captures the local features (noisy trends) present in the identification data instead of the desired global characteristics (useful system dynamics) [34]. In this work, it is demonstrated that by carefully choosing the network's (model's) complexity, which is suitable enough to capture the useful desired dynamics only, the problem of identifying unstable delayed systems can be performed. Furthermore, the accuracy of estimated parameters can be enhanced based on a guaranteed convergence criterion, which is demonstrated in the simulation studies.

The manuscript is organized as follows: In next Sect. 2, the related works, problems and the objectives of proposed work have been discussed. In Sect. 3, the concept, challenges, and advantages of closed-loop identification have been discussed. In Sect. 4, the formulation of proposed technique has been presented. The Sect. 5 is devoted for the detailed development of proposed identification algorithm. The results are included in Sect. 6. The manuscript is concluded with possible future directions in Sect. 7.

2 Related Works, Problems and Objectives

In recent studies, it is shown that the weights of neural networks can also be related to the actual system parameters [35–37]. In this context, some modeling and identification approaches are developed, where it is demonstrated that by using some reduced or simplified neural network architectures parametric identification can be performed.

The work presented by Ho et al. [38] uses artificial neural networks to identify unknown systems as first order plus time delay models. The authors demonstrated that the controller tuned based on the identified model performed better than the conventional methods. However, the limitation of this method is its limited scope where only stable delayed process dynamics can be identified. In [39], authors performed a neural network based technique for the identification of stable-FOD (first-order delayed) processes. However, the main drawbacks of this method is that only stable systems can be estimated and the identified parameters are biased in the presence of measurement noise.

Other useful contributions can be found in [40, 41], where discrete transfer function models are identified in terms of trained neural network weights. However, this method is not suitable for time delay estimation. Chon and Cohen [42] presented an approach to estimate time-series model using trained network employing polynomial activation. However, the choice of polynomial orders is critical for accurate estimation.

In [43], a linear recurrent neural is proposed for identifying delay-free systems. In this approach, they demonstrated that the distributed parallel neural structure can be used for identifying multiple transfer functions together. However,

this approach cannot be used for identifying unstable systems having time delay. A simplified block-oriented RNN is designed in [44], to extract the delay-free, stable Wiener model, directly in the form of trained network weights. It is also demonstrated that a neural proportional, integral, and derivative (PID) controller designed using this identified model gives better performance on the actual system as compared to the conventional PID controller.

The above discussions shows that the neural networks are capable to interpret parametric models and can outperform the conventional techniques. However, in the existing literature the parametric identifiability of RNNs are limited to stable delayed dynamics only, where the time delay estimation is not efficient. This was the main motivation behind the proposed approach to utilize RNNs for parametric identification of unstable time delayed systems, which is more challenging and has not been reported yet. Hence, in this work, a new method is proposed to identify an unknown unstable delayed dynamics of any order to its equivalent unstable first- or second-order delayed (FOD or SOD) model for improved controller design and model analysis purposes. Also, where all the model parameters as: gain, time constants, and time delay can be derived in terms of the neural network weights.

The conventional contributions that deals with the problem of unstable delayed process identification are also reported in literature. The reaction curve based methods are developed in [45–47] for unstable-FOD (UFOD) process modeling, which is simple, but the time delay estimation is inaccurate. In [48], a closed-loop step response method is developed. However, a suitable choice of step length and damping factor is needed for convergence. A curve fitting approach is proposed in [49] to estimate unstable systems with delay. However, this approach is practically limited as the rational model is needed in advance.

The relay feedback methods are also found to be useful for estimating unstable models with time delay [50–58]. However, their success rely upon the existence of limit cycles, which is hard to get for unstable systems. Furthermore, in most of the relay-based techniques, the limit cycles corrupted by measurement noise are required to be cleaned first by using some Fourier series based curve fitting methods before applying the relay identification techniques to it [52, 54]. However, the proposed work does not impose any limiting conditions on parameters, and the noisy data can be directly used for model identification.

More literature works that deal with unstable system identification are included here. The reference [21], talks about the modified Box–Jenkins models for unstable systems. In [59, 60], a virtual control method was developed to recover model of an unstable system through closed-loop data. In [61], an impulse response approach was adapted for estimating unstable dynamics. However, in all these references, the

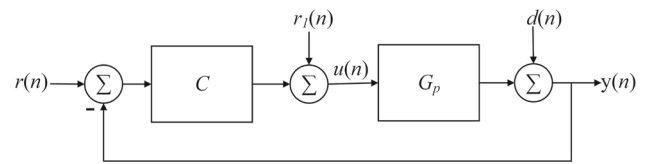


Fig. 1 Closed-loop identification system

time delays are not included in the model structure, which makes it relatively simpler problem.

From these existing contributions, one can infer that the estimation of unstable dynamics with time delay is a challenging area of research. Also, due to various issues like plant's safety, predictor's stability, input's correlation and nonlinear optimization, the problem of identifying unstable delayed systems is less reported in the literature. It is also discussed that the neural network with its various attractive features can be a better alternative in solving such problems.

This work proposed a novel closed-loop identification approach, where the closed-loop data (which is uncorrelated to noise disturbance) and the controller information (which is used to ensure predictor's stability) is utilized in proposed CLDRNN architecture, to identify the unknown unstable delayed process. The developed CLDRNN structure is fully parametric and transparent to carry out the closed-loop identification of unstable time delay models with more accuracy and guaranteed convergence.

The main objectives and contributions of this work are outlined as:

1. An RNN based, closed-loop identification technique is proposed to identify the unstable process models with time delay, which can also handle controller complexity and long process time delays.
2. The mathematical relationships are formulated to get model and time delay parameters using CLDRNN weights.
3. Theorems are proved to obtain the theoretical conditions on convergence and adaptive learning for the proposed algorithm, along with the Monte-Carlo experiments to show the convergence for practical examples.
4. The convergence and parameter initialization issues are discussed under model uncertainty and noisy datasets.
5. Extensive simulations are performed to demonstrate that the proposed method can produce robust and unbiased parameter estimates with excellent convergence.

3 Closed-Loop Identification

Identification in closed-loop has gained a lot of attention in recent years as most of the industrial systems operate in closed-loop [22, 34, 62]. Also, the unstable systems cannot

operated in open-loop because of safety issues. Further, it has been observed that, for model-based control relevant applications the closed-loop configuration is often considered to be the optimal experimental setup [59, 63]. Additionally, identifying the closed-loop behavior of the unstable systems is important as they are ultimately going to be operated in a closed-loop only [21].

A closed-loop identification system is presented in Fig. 1, where n is discrete time the reference signal $r(n)$ and an additional signal $r_1(n)$ act as inputs at set-point and at controller's C output. The input, output and disturbance signals for the process G_p are defined as $u(n)$, $y(n)$ and $d(n)$, respectively.

The direct, indirect and joint approaches are popular for closed-loop identification [34, 62]. In the direct approach, the open-loop identification techniques can be used directly by using input $u(n)$ and output $y(n)$ data. The advantage of this technique is that the knowledge of feedback mechanism and controller is not required. Still, the limitation is that the biased estimates are obtained in the presence of measurement noise due to the correlation between input $u(n)$ and noise disturbance $d(n)$ [60]. One more disadvantage of a direct approach is that the user has little control over input design because the controller derives the process. Therefore, it is not easy to design inputs, which can produce persistent excitation and information-rich data [34]. It is mentioned in [21] that, without ensuring the predictor's stability the direct closed-loop identification can not be applied for unstable systems.

In the indirect approach, the overall system relating reference input $r(n)$ to output $y(n)$ is identified first. Then the process model for G_p is recovered from the overall closed-loop system with the knowledge of the controller C . The main benefits of this approach are uncorrelated input, stable predictor and any open-loop identification techniques can be used to identify the overall closed-loop system [22]. The major drawback of the indirect method is that the recovery of an equivalent model for process G_p becomes difficult and requires further model reductions [62].

The joint input–output approach uses the identification of two separate systems by relating set-point input $r(n)$ with output $y(n)$ and input $u(n)$, respectively. Then, the equivalent process model for G_p can be recovered by dividing the first system to the second system. The advantage of this approach is that the controller is not required to be known. Still, the drawback associated with this technique is that the denominator parts (poles) of both identified systems should be identical for accurate estimation [34, 62].

The objective of this manuscript is to perform improved closed-loop identification, especially for unstable processes with time delays and to overcome the limitations associated with the closed-loop identification approaches, as pointed out in the above discussion. The proposed technique is presented in the next section, which can produce unbiased parameter estimates in a noisy environment, with an added advantage

to directly recover an unknown unstable delayed dynamics in terms of UFOD and USOD models.

The proposed method has following assumptions for performing identification experiment:

A1: The data generation for identification is performed in closed-loop with a known controller.

A2: The reference input is piecewise constant and persistently exciting to get information-rich data.

A3: The initial conditions are taken as zero for both input as well as output signals.

A4: A zero mean and finite variance white noise is used as disturbance signal.

Remark 1 The above assumptions are the standard considerations which are adopted in most of the identification techniques. The assumption in A1 is essential to perform identification of unstable systems in closed-loop. Otherwise, in open-loop, the output would be unbounded resulting in safety concerns of the plants. Also, in cases where the controllers are not known the joint input–output approach can be employed. Furthermore, the applicability of proposed approach method will still be valid if the assumptions in A2, A3 & A4 are relaxed. However, in that situation it might have an impact on convergence, accuracy and consistency of the estimated parameters.

4 Proposed CLDRNN Identification Approach

This section is devoted in developing the following novel concepts of the proposed work.

1. A novel CLDRNN architecture is proposed to enable closed-loop identification.
2. The unique thing about CLDRNN is that the time delayed process model can be directly extracted from its network weights.
3. New mathematical formulation has been developed in Theorems 1 and 2 for computing unstable delayed model parameters.
4. The adaptive learning rate criterion for CLDRNN training is obtained by Lyapunov theory in Theorem 1.
5. An original condition has been derived for fractional delay, which ensures the guaranteed convergence of the proposed algorithm.

The proposed CLDRNN based closed-loop identification and modeling scheme is shown in Fig. 2, where it can be observed that the CLDRNN mimics the complete closed-loop system. The CLDRNN consists of three layers. The first layer referred to the error layer, which calculates the error signal $e(n)$ between reference input $r(n)$ and estimated output $\hat{y}(n)$. The second layer is called the controller layer, whose

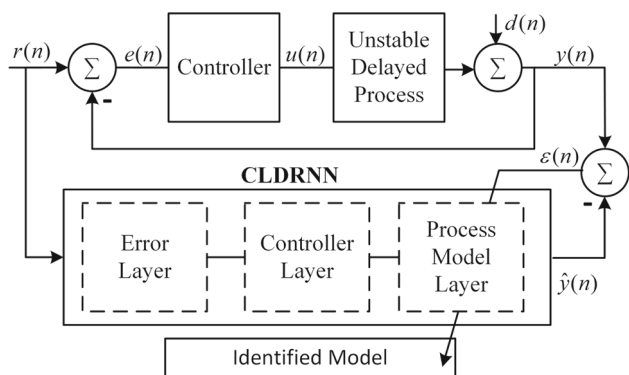


Fig. 2 The proposed closed-loop identification and modeling scheme using CLDRNN

structure and weights are selected according to a known controller, which is used to stabilize the actual unstable process. The last layer is referred to as the process layer, which is responsible for capturing the unknown unstable delayed process dynamics in terms of its optimized weights.

The proposed CLDRNN is trained with closed-loop data, and during training weights of error and controller layer are kept fixed. In contrast, weights of process layer are altered to reduce the difference error $\varepsilon(n)$ between actual output $y(n)$ and estimated output $\hat{y}(n)$. Finally, the process model can be recovered directly through the process layer weights of CLDRNN as suitable UFOD or USOD process models.

A more detailed CLDRNN architecture, utilizing the parallel connection mode, is shown in Fig. 3, which includes three layers whose connection weights are w and activation functions denoted by $f(\Sigma)$. The proposed CLDRNN contains all the components of a typical closed-loop system, i.e., error computation part, controller part, and the process identifier part within its architecture. The advantage of the CLDRNN structure is that it is trained with closed-loop data to directly identify the unknown unstable delayed dynamics in terms of UFOD and USOD model parameters. In the following subsection, a method is demonstrated, to include a PID-type controller in the controller layer of the CLDRNN.

4.1 Inclusion of the PID-Type Controller in the CLDRNN

It is clearly mentioned in the literature [22, 34, 62] that a pre-stabilizing controller of some kind is essentially required for the identification of unstable system dynamics. Also, the identification experiment for unstable processes cannot be performed, without ensuring the stability of the overall closed-loop system [21, 48]. The authors in [59, 60] recommended the use of stabilizing controller as virtual controller to demonstrate its utility in identifying unstable systems. In general, the PID-type (P, PI, PD, or PID) controllers are commonly employed to stabilize the unstable dynamics in order

to generate bounded data for identification [36]. In the proposed approach, this controller information has been utilized in the CLDRNN structure as virtual controller [59, 60], to gain the following advantages:

1. It helps the CLDRNN structure to mimic the complete closed-loop system, from which the equivalent models (UFOD or USOD) of actual unstable delayed processes can be directly extracted in terms of the process model layer weights.
2. It helps in utilizing the reference (set point) input $r(n)$ directly for identification, which is uncorrelated to the noise disturbances $d(n)$.
3. It helps in keeping the CLDRNN structure stable (generates stable output $\hat{y}(n)$ predictions) during training, as we are identifying unstable systems.

In the CLDRNN architecture of Fig. 3, for error calculation, the reference signal $r(n)$ is subtracted from estimated output $\hat{y}(n)$ by setting the weights $w_0^e = 1$ and $w_1^e = -1$. To set the weights in the second layer of the CLDRNN, consider the parallel form of PID controller in terms of unit sample delay (USD) operator z^{-1} with uniform sampling time T_s as

$$C(z^{-1}) = K_p + \frac{K_i T_s z^{-1}}{1 - z^{-1}} + \frac{K_d \beta (1 - z^{-1})}{1 - (1 - \beta T_s) z^{-1}}, \tag{1}$$

where K_p is proportional gain, K_i is integral gain, K_d is derivative gain, and β represents the filter coefficient of the derivative term. The PID controller presented in (1), is popular and extensively used in various industrial applications. The reason is its simplicity and easy implementation on digital hardware in almost all computer-aided controllers, with the addition of hold device such as the zero-order hold (ZOH), to generate analog control signals for the actual process. If the error signal $e(n)$ is the input and $u(n)$ is the output of the CLDRNN's controller layer then according to (1), one can have the following difference equation representation as

$$\begin{aligned} u(n) = & (2 - \beta T_s)u(n - 1) + (\beta T_s - 1)u(n - 2) \\ & + (K_p + K_d \beta)e(n) + (K_i T_s - K_p(2 - \beta T_s) \\ & - 2K_d \beta)e(n - 1) + (K_d \beta + K_p(1 - \beta T_s) \\ & - K_i T_s(1 - \beta T_s))e(n - 2), \end{aligned} \tag{2}$$

where the weights of the controller layer of CLDRNN are computed by using known controller parameter values in (2)

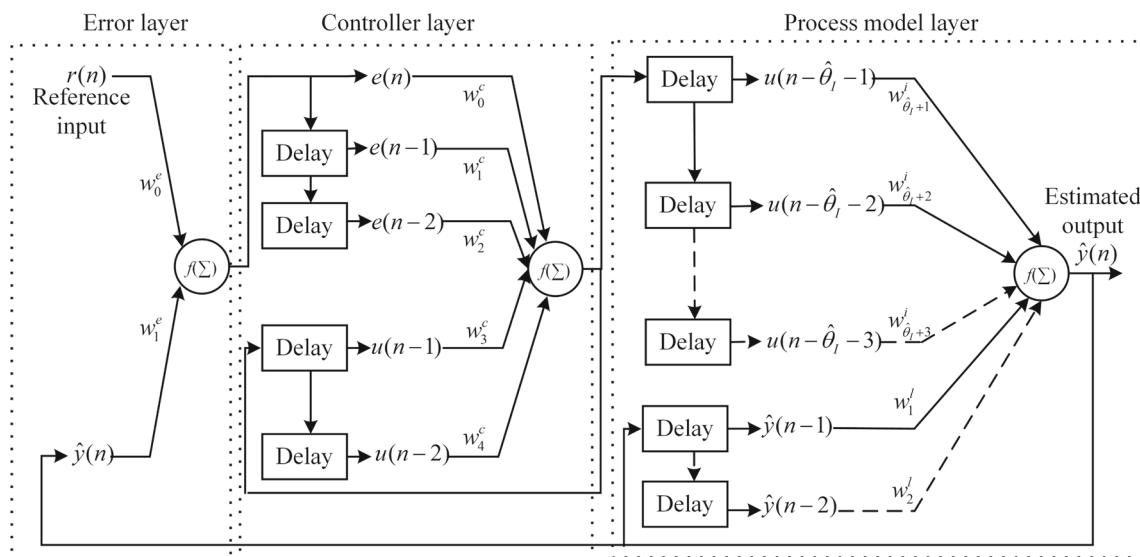


Fig. 3 The proposed CLDRNN architecture for the closed-loop process identification as UFOD and USOD models. Delay represents unit sample delay. Note that, the terms with dashed lines are used for USOD system modeling

as:

$$w_0^c = K_p + K_d\beta, \tag{3}$$

$$w_1^c = K_iT_s - K_p(2 - \beta T_s) - 2K_d\beta, \tag{4}$$

$$w_2^c = K_d\beta + K_p(1 - \beta T_s) - K_iT_s(1 - \beta T_s), \tag{5}$$

$$w_3^c = 2 - \beta T_s, \tag{6}$$

$$w_4^c = \beta T_s - 1. \tag{7}$$

Therefore, for any combinations like P, PI, PD, or PID, the controller layer weights of the CLDRNN can be calculated by using expressions from (3) to (7).

Remark 2 It is also possible to consider a more sophisticated PID-type controller structure that includes nonlinearities, integral windups, and saturation terms in the controller layer of the proposed CLDRNN, by incorporating separate training of controller layer with suitable activation functions [23, 44]. However, to keep the analysis fully parametric, we have restricted the proposed work to a PID controller of type (1) with a linear activation function only.

The following subsections used to develop a new methodology to obtain the parameters of UFOD and USOD system models in terms of the CLDRNN weights. The main attraction of the developed mathematical formulation is that the integer and non-integer (fractional) parts of the time delay can also be computed along with other rational parameters, directly from the process model layer of the CLDRNN.

4.2 UFOD System Identification Using CLDRNN

Let the UFOD system models $G(s)$ to be identified is represented as:

$$\hat{G}(s) = \frac{\hat{K}e^{-\hat{\theta}s}}{\hat{\tau}s - 1}, \tag{8}$$

where \hat{K} is the gain, $\hat{\theta}$ is the time delay and $\hat{\tau}$ is the time constant of the identified UFOD model. In the proposed technique the unknowns (\hat{K} , $\hat{\theta}$ and $\hat{\tau}$) of (8) are determined through closed-loop training of the CLDRNN. Considering that the time delay $\hat{\theta}$ is expressed in discrete domain as the combination of integer $\hat{\theta}_I$ and fractional $\hat{\theta}_f$; $-1 < \hat{\theta}_f < 1$, delay terms as $\hat{\theta} = (\hat{\theta}_I + \hat{\theta}_f) T_s$, then for UFOD modeling the typical estimated output of the process model layer of the CLDRNN is given by:

$$\hat{y}(n) = w_1^l \hat{y}(n-1) + w_{\hat{\theta}_I+1}^i u(n - \hat{\theta}_I - 1) + w_{\hat{\theta}_I+2}^i u(n - \hat{\theta}_I - 2), \tag{9}$$

where the terms w_1^l , $w_{\hat{\theta}_I+1}^i$ and $w_{\hat{\theta}_I+2}^i$ in (9) are the associated weights of $\hat{y}(n-1)$, $u(n - \hat{\theta}_I - 1)$ and $u(n - \hat{\theta}_I - 2)$, respectively. Now by using the USD operator z^{-1} such that $z^{-1}u(n) = u(n-1)$ in (9), one can have an equivalent discrete-time transfer function model corresponding to the continuous-time UFOD model of (8) as:

$$\hat{G}(z^{-1}) = \frac{w_{\hat{\theta}_I+1}^i z^{-1} + w_{\hat{\theta}_I+2}^i z^{-2}}{1 - w_1^l z^{-1}} z^{-\hat{\theta}_I}. \tag{10}$$

To recover the model in (8), in terms of the trained CLDRNN’s process layer weights present in (10), a ZOH based modified \mathcal{Z} -transform [34, 64, 65] approach is utilized, which is written and proved in *Theorem 1* outlined as:

Theorem 1 Consider the modified \mathcal{Z} -transform of UFOD system with ZOH = $(1 - e^{-sT_s})/s$, to have the following relations as:

$$\begin{aligned} &\mathcal{Z} \left[\mathcal{L}^{-1} \left\{ \frac{(1 - e^{-sT_s}) \hat{K} e^{-s(\hat{\theta}_l + \hat{\theta}_f)T_s}}{s(\hat{\tau}s - 1)} \right\} \right]; -1 < \hat{\theta}_f < 1 \\ &= \frac{w_{\hat{\theta}_l+1}^i z^{-1} + w_{\hat{\theta}_l+2}^i z^{-2}}{1 - w_1^l z^{-1}} z^{-\hat{\theta}_l}, \text{ where} \\ &\hat{K} = (w_{\hat{\theta}_l+1}^i + w_{\hat{\theta}_l+2}^i) / (w_1^l - 1), \hat{\tau} = T_s / \log(w_1^l), \\ &\hat{\theta}_f = 1 - \log \left(\frac{w_{\hat{\theta}_l+2}^i + w_{\hat{\theta}_l+1}^i w_1^l}{w_{\hat{\theta}_l+1}^i + w_{\hat{\theta}_l+2}^i} \right) / \log(w_1^l), \hat{\theta} = (\hat{\theta}_l + \hat{\theta}_f)T_s. \end{aligned} \tag{11}$$

Proof Let us consider the left hand side term of the *Theorem*, with substitution $z^{-1} = e^{-sT_s}$ as

$$\hat{G}(z^{-1}) = (1 - z^{-1})z^{-\hat{\theta}_l} \mathcal{Z} \left[\mathcal{L}^{-1} \left\{ \frac{\hat{K}}{s(\hat{\tau}s - 1)} e^{-s\hat{\theta}_f T_s} \right\} \right], \tag{12}$$

where denoted by \mathcal{L}^{-1} is the inverse Laplace transform. Now by using partial fraction computation in (12), to get

$$\begin{aligned} \hat{G}(z^{-1}) &= (1 - z^{-1})z^{-\hat{\theta}_l} \times \\ &\mathcal{Z} \left[\mathcal{L}^{-1} \left\{ \frac{\hat{K} \hat{\tau} e^{-s\hat{\theta}_f T_s}}{\hat{\tau}s - 1} \right\} - \mathcal{L}^{-1} \left\{ \frac{\hat{K} e^{-s\hat{\theta}_f T_s}}{s} \right\} \right]. \end{aligned} \tag{13}$$

The subsequent *Lemma* and *Corollary* are proved to simplify the expression in (13) as:

Lemma 1 For the following unstable transfer function the modified \mathcal{Z} -transform is computed as:

$$\mathcal{Z} \left[\mathcal{L}^{-1} \left\{ \frac{\hat{\tau} e^{-s\hat{\theta}_f T_s}}{\hat{\tau}s - 1} \right\} \right]; -1 < \hat{\theta}_f < 1 = \frac{e^{-(\hat{\theta}_f-1)T_s/\hat{\tau}} z^{-1}}{1 - e^{T_s/\hat{\tau}} z^{-1}}. \tag{14}$$

Proof Considering the LHS of (14) with the time-shifting property of Laplace transform, to get:

$$\mathcal{Z} \left[\mathcal{L}^{-1} \left\{ \frac{\hat{\tau} e^{-s\hat{\theta}_f T_s}}{\hat{\tau}s - 1} \right\} \right] = \mathcal{Z} \left[e^{(t-\hat{\theta}_f T_s)/\hat{\tau}} 1(t - \hat{\theta}_f T_s) \right], \tag{15}$$

where $1(t - \hat{\theta}_f T_s)$ represents an unit step signal delayed by $\hat{\theta}_f T_s$.

Now, computing the sum of geometric progression with infinite terms and $|z| > \left| e^{T_s/\hat{\tau}} \right|$, to have

$$e^{-\hat{\theta}_f T_s/\hat{\tau}} \sum_{n=1}^{\infty} e^{nT_s/\hat{\tau}} z^{-n} = \frac{e^{-(\hat{\theta}_f-1)T_s/\hat{\tau}} z^{-1}}{1 - e^{T_s/\hat{\tau}} z^{-1}}. \tag{16}$$

Corollary 1 For a delayed integral transfer function the modified \mathcal{Z} -transform is given as:

$$\mathcal{Z} \left[\mathcal{L}^{-1} \left\{ \frac{1}{s} e^{-s\hat{\theta}_f T_s} \right\} \right]; -1 < \hat{\theta}_f < 1 = \frac{z^{-1}}{1 - z^{-1}}. \tag{17}$$

Proof To proof this, do the substitution of $1/\hat{\tau} = 0$, in (14), to get

$$\mathcal{Z} \left[\mathcal{L}^{-1} \left\{ \frac{1}{s} e^{-s\hat{\theta}_f T_s} \right\} \right] = \frac{z^{-1}}{(1 - z^{-1})}. \tag{18}$$

Now, applying the outcomes of *Lemma 1*, and *Corollary 1*, in (13), to get

$$\hat{G}(z^{-1}) = (1 - z^{-1})z^{-\hat{\theta}_l} \hat{K} \left[\frac{e^{-(\hat{\theta}_f-1)T_s/\hat{\tau}} z^{-1}}{1 - e^{T_s/\hat{\tau}} z^{-1}} - \frac{z^{-1}}{1 - z^{-1}} \right] \tag{19}$$

on simplifying (19), one can have

$$\begin{aligned} \hat{G}(z^{-1}) &= \hat{K} \frac{(e^{-(\hat{\theta}_f-1)T_s/\hat{\tau}} - 1) z^{-1}}{1 - e^{T_s/\hat{\tau}} z^{-1}} z^{-\hat{\theta}_l} \\ &+ \hat{K} \frac{(e^{T_s/\hat{\tau}} - e^{-(\hat{\theta}_f-1)T_s/\hat{\tau}}) z^{-2}}{1 - e^{T_s/\hat{\tau}} z^{-1}} z^{-\hat{\theta}_l} \end{aligned} \tag{20}$$

if in (20), each individual z^{-1} power coefficients are assigned in terms of the CLDRNN’s process model layer weights, to have

$$\hat{G}(z^{-1}) = \frac{w_{\hat{\theta}_l+1}^i z^{-1} + w_{\hat{\theta}_l+2}^i z^{-2}}{1 - w_1^l z^{-1}} z^{-\hat{\theta}_l}, \tag{21}$$

where the coefficients of (21) and (20) are related by

$$w_{\hat{\theta}_l+1}^i = \hat{K} (e^{-(\hat{\theta}_f-1)T_s/\hat{\tau}} - 1), \tag{22}$$

$$w_{\hat{\theta}_l+2}^i = \hat{K} (e^{T_s/\hat{\tau}} - e^{-(\hat{\theta}_f-1)T_s/\hat{\tau}}), \tag{23}$$

and

$$w_1^l = e^{T_s/\hat{\tau}}. \tag{24}$$

The model parameters of UFOD system in (8) can be computed from (22), (23) and (24) as:

$$\hat{K} = (w_{\hat{\theta}_I+1}^i + w_{\hat{\theta}_I+2}^i) / (w_1^i - 1), \quad (25)$$

$$\hat{\tau} = T_s / \log(w_1^i), \quad (26)$$

$$\hat{\theta}_f = 1 - \log\left(\frac{w_{\hat{\theta}_I+2}^i + w_{\hat{\theta}_I+1}^i w_1^i}{w_{\hat{\theta}_I+1}^i + w_{\hat{\theta}_I+2}^i}\right) / \log(w_1^i), \quad (27)$$

and

$$\hat{\theta} = (\hat{\theta}_I + \hat{\theta}_f)T_s. \quad (28)$$

Remark 3 Note that, the parameters in the expressions of (25), (26) and (27) can easily be recovered from optimized CLDRNN weights. However, the parameter $\hat{\theta}_I$ is estimated by an iterative algorithm, whose details are presented in the next section.

4.3 USOD System Identification Using CLDRNN

Now, consider the case when an unknown unstable process dynamics identified as the following USOD process model given by:

$$\hat{G}(s) = \frac{\hat{K}e^{-\hat{\theta}s}}{(\hat{\tau}_1s - 1)(\hat{\tau}_2s + 1)}, \quad (29)$$

where the notations \hat{K} , $\hat{\theta}$ and $\hat{\tau}_1$ & $\hat{\tau}_2$ are used to represent gain, time delay and time constants of the model. The typical CLDRNN's output response for identifying systems as USOD model in (29) is given by

$$\begin{aligned} \hat{y}(n) = & w_1^i \hat{y}(n-1) + w_2^i \hat{y}(n-2) + w_{\hat{\theta}_I+1}^i u(n - \hat{\theta}_I - 1) \\ & + w_{\hat{\theta}_I+2}^i u(n - \hat{\theta}_I - 2) + w_{\hat{\theta}_I+3}^i u(n - \hat{\theta}_I - 3), \end{aligned} \quad (30)$$

rewrite the expression in (30), to its equivalent discrete time-delayed transfer function form as:

$$\hat{G}(z^{-1}) = \frac{w_{\hat{\theta}_I+1}^i z^{-1} + w_{\hat{\theta}_I+2}^i z^{-2} + w_{\hat{\theta}_I+3}^i z^{-3}}{(1 - \alpha_1 z^{-1})(1 - \alpha_2 z^{-1})} z^{-\hat{\theta}_I}, \quad (31)$$

where the terms α_1 and α_2 in (31) are defined as:

$$\alpha_1 = 0.5(w_1^i + ((w_1^i)^2 + 4w_2^i)^{0.5}), \quad (32)$$

$$\alpha_2 = 0.5(w_1^i - ((w_1^i)^2 + 4w_2^i)^{0.5}). \quad (33)$$

The identification formulation of USOD model of (29), in terms of the CLDRNN process layer weights is presented as following:

Theorem 2 Consider the modified \mathcal{Z} -transform of USOD system as:

$$\begin{aligned} \mathcal{Z} \left[\mathcal{L}^{-1} \left\{ \frac{(1 - e^{-sT_s})}{s} \frac{\hat{K}}{(\hat{\tau}_1s - 1)(\hat{\tau}_2s + 1)} e^{-s(\hat{\theta}_I + \hat{\theta}_f)T_s} \right\} \right]; \\ -1 < \hat{\theta}_f < 1 = \frac{w_{\hat{\theta}_I+1}^i z^{-1} + w_{\hat{\theta}_I+2}^i z^{-2} + w_{\hat{\theta}_I+3}^i z^{-3}}{(1 - \alpha_1 z^{-1})(1 - \alpha_2 z^{-1})} z^{-\hat{\theta}_I}, \end{aligned}$$

with α_1 in (32) and α_2 in (33), then

$$\begin{aligned} \hat{K} = \frac{w_{\hat{\theta}_I+1}^i + w_{\hat{\theta}_I+2}^i + w_{\hat{\theta}_I+3}^i}{(\alpha_1 - 1)(1 - \alpha_2)}, \hat{\tau}_1 = \frac{T_s}{\log(\alpha_1)}, \hat{\tau}_2 = -\frac{T_s}{\log(\alpha_2)}, \\ \hat{\theta}_f = \frac{\hat{\tau}_1}{T_s} \log \left(\frac{\hat{K} \hat{\tau}_1 (\alpha_2 - \alpha_1)}{(\hat{\tau}_1 + \hat{\tau}_2) [w_{\hat{\theta}_I+3}^i - w_{\hat{\theta}_I+1}^i \alpha_1 + \hat{K} \alpha_1 (\alpha_2 - 1)]} \right), \quad (34) \\ \hat{\theta} = (\hat{\theta}_I + \hat{\theta}_f)T_s. \end{aligned}$$

Proof Let us consider the LHS term of *Theorem 2*, by replacing z^{-1} by e^{-sT_s} , to get the expression

$$\begin{aligned} \hat{G}(z^{-1}) = (1 - z^{-1})z^{-\hat{\theta}_I} \times \\ \mathcal{Z} \left[\mathcal{L}^{-1} \left\{ \frac{\hat{K}}{s(\hat{\tau}_1s - 1)(\hat{\tau}_2s + 1)} e^{-s\hat{\theta}_f T_s} \right\} \right], \end{aligned} \quad (35)$$

on employing partial fraction computation in (35), to get

$$\begin{aligned} \hat{G}(z^{-1}) = \frac{(1 - z^{-1})z^{-\hat{\theta}_I} \hat{K}}{(\hat{\tau}_1 + \hat{\tau}_2)} \times \\ \mathcal{Z} \left[\mathcal{L}^{-1} \left\{ \left(\frac{\hat{\tau}_1^2}{(\hat{\tau}_1s - 1)} + \frac{\hat{\tau}_2^2}{(\hat{\tau}_2s + 1)} - \frac{(\hat{\tau}_1 + \hat{\tau}_2)}{s} \right) e^{-s\hat{\theta}_f T_s} \right\} \right] \end{aligned} \quad (36)$$

by applying *Lemma 1* and *Corollary 1*, in (36), to compute the following terms:

$$\begin{aligned} \mathcal{Z} \left[\mathcal{L}^{-1} \left\{ \frac{\hat{\tau}_1}{\hat{\tau}_1s - 1} e^{-s\hat{\theta}_f T_s} \right\} \right] \\ = \frac{e^{-(\hat{\theta}_f - 1)T_s / \hat{\tau}_1} z^{-1}}{1 - e^{T_s / \hat{\tau}_1} z^{-1}}; -1 < \hat{\theta}_f < 1, \end{aligned} \quad (37)$$

$$\begin{aligned} \mathcal{Z} \left[\mathcal{L}^{-1} \left\{ \frac{\hat{\tau}_2}{\hat{\tau}_2s + 1} e^{-s\hat{\theta}_f T_s} \right\} \right] \\ = \frac{e^{(\hat{\theta}_f - 1)T_s / \hat{\tau}_2} z^{-1}}{1 - e^{-T_s / \hat{\tau}_2} z^{-1}}; -1 < \hat{\theta}_f < 1, \end{aligned} \quad (38)$$

and

$$\mathcal{Z} \left[\mathcal{L}^{-1} \left\{ \frac{\hat{K}}{s} e^{-s\hat{\theta}_f T_s} \right\} \right] = \frac{\hat{K} z^{-1}}{1 - z^{-1}}; -1 < \hat{\theta}_f < 1, \quad (39)$$



on substituting (37), (38) and (39) into (36), then after simplification and collecting different z^{-1} power coefficients terms, to have

$$\hat{G}(z^{-1}) = \frac{w_{\hat{\theta}_I+1}^i z^{-1} + w_{\hat{\theta}_I+2}^i z^{-2} + w_{\hat{\theta}_I+3}^i z^{-3}}{(1 - \alpha_1 z^{-1})(1 - \alpha_2 z^{-1})} z^{-\hat{\theta}_I}, \quad (40)$$

where the coefficient terms are given by

$$w_{\hat{\theta}_I+1}^i = \hat{K} \left[\frac{\hat{\tau}_1 e^{-(\hat{\theta}_f-1)T_s/\hat{\tau}_1} + \hat{\tau}_2 e^{(\hat{\theta}_f-1)T_s/\hat{\tau}_2}}{\hat{\tau}_1 + \hat{\tau}_2} - 1 \right], \quad (41)$$

$$w_{\hat{\theta}_I+2}^i = \hat{K} \left(e^{T_s/\hat{\tau}_1} + e^{-T_s/\hat{\tau}_2} \right) - \frac{\hat{K}}{(\hat{\tau}_1 + \hat{\tau}_2)} \left[\hat{\tau}_1 e^{-(\hat{\theta}_f-1)T_s/\hat{\tau}_1} (1 + e^{-T_s/\hat{\tau}_2}) \right] - \frac{\hat{K}}{(\hat{\tau}_1 + \hat{\tau}_2)} \left[\hat{\tau}_1 e^{(\hat{\theta}_f-1)T_s/\hat{\tau}_2} (1 + e^{T_s/\hat{\tau}_1}) \right], \quad (42)$$

$$w_{\hat{\theta}_I+3}^i = \frac{\hat{K}}{(\hat{\tau}_1 + \hat{\tau}_2)} \left[\hat{\tau}_1 e^{-(\hat{\theta}_f-1)T_s/\hat{\tau}_1} e^{-T_s/\hat{\tau}_2} \right] + \frac{\hat{K}}{(\hat{\tau}_1 + \hat{\tau}_2)} \left[\hat{\tau}_2 e^{(\hat{\theta}_f-1)T_s/\hat{\tau}_2} e^{T_s/\hat{\tau}_1} \right] - \hat{K} e^{T_s/\hat{\tau}_1} e^{-T_s/\hat{\tau}_2}, \quad (43)$$

and

$$\alpha_1 = e^{T_s/\hat{\tau}_1}, \alpha_2 = e^{-T_s/\hat{\tau}_2}. \quad (44)$$

Now, add (41), (42) and (43) with substitution from (44), to have

$$\hat{K} = \frac{w_{\hat{\theta}_I+1}^i + w_{\hat{\theta}_I+2}^i + w_{\hat{\theta}_I+3}^i}{(\alpha_1 - 1)(1 - \alpha_2)}, \quad (45)$$

from (44), one can have

$$\hat{\tau}_1 = T_s/\log(\alpha_1), \hat{\tau}_2 = -T_s/\log(\alpha_2), \quad (46)$$

on multiplying (41), with $e^{T_s/\hat{\tau}_1}$ and subtracting it from (43), with substitutions from (44), to have

$$\hat{\theta}_f = \frac{\hat{\tau}_1}{T_s} \log \left(\frac{\hat{K} \hat{\tau}_1 \alpha_1 (\alpha_2 - \alpha_1)}{(\hat{\tau}_1 + \hat{\tau}_2) \left[w_{\hat{\theta}_I+3}^i - w_{\hat{\theta}_I+1}^i \alpha_1 + \hat{K} \alpha_1 (\alpha_2 - 1) \right]} \right), \quad (47)$$

also, the estimated time delay parameter $\hat{\theta}$ is computed by $\hat{\theta} = (\hat{\theta}_I + \hat{\theta}_f)T_s$, as in (28).

Based on the formulation done in this section, a guaranteed convergence condition has been proposed and an iterative learning algorithm is developed in following section

to solve the nonlinear optimization problem for identifying time-delayed systems.

5 Guaranteed Convergence Condition and Learning of CLDRNN

On the basis of the formulations done in the previous section, it can be inferred that the guaranteed convergence for obtaining optimal model parameters can only be obtained if the following condition is satisfied as:

$$-1 < \hat{\theta}_f < 1. \quad (48)$$

Hence, by monitoring a single parameter value only, one can ensure the optimal model estimation and algorithms convergence at the same time. Also, for identifying the time delay $\hat{\theta}$, there is no direct formula for computing the integer part $\hat{\theta}_I$; therefore, it is recommended to extract it from the non-integer part iteratively until the convergence condition given in (48) is satisfied.

The proposed iterative process for CLDRNN training is begin with the initial choice of the parameter $\hat{\theta}_I = 0$, where an arbitrary choice of integer time delay can also be considered based on some intuitive knowledge. After each iteration, the following expression is used to update the integer part $\hat{\theta}_I$ from the estimated fractional part $\hat{\theta}_f$ as:

$$\hat{\theta}_I(\text{present}) = \hat{\theta}_I(\text{previous}) + \hat{\theta}_f(\text{present}) - \text{rem} \left\{ \hat{\theta}_f(\text{present}), 1 \right\}, \quad (49)$$

the updated value $\hat{\theta}_I(\text{present})$ is then used in next iteration and the same updation is repeated until the guaranteed global convergence condition $-1 < \hat{\theta}_f < 1$ is satisfied.

Remark 4 Note that, the approach discussed above plays an instrumental role in estimating unbiased parameters and in achieving global convergence even if the measurement noise severely corrupts the identification data or the modeling uncertainties are present. The reason behind this is the identification of $\hat{\theta}_f$, that enables an accurate estimate of $\hat{\theta}_I$, which then further improves the identification of the other model parameters as well.

Let the cost function F_k at k th (used as subscript k in all subsequent representations) training epoch is minimized to train the proposed CLDRNN as:

$$F_k = \frac{1}{2N} \sum_{n=1}^N \varepsilon_k^2(n). \quad (50)$$

where N denotes the number of training samples and $\varepsilon_k(n)$ is the difference error at k th epoch, computed by:

$$\varepsilon_k(n) = y(n) - \hat{y}_k(n). \quad (51)$$

The terms $y(n)$ and $\hat{y}_k(n)$ in (51) are true and identified model outputs. If all errors are accumulated in a single difference error vector \mathcal{E}_k as:

$$\mathcal{E}_k = [\varepsilon_k(1) \ \varepsilon_k(2) \ \dots \ \varepsilon_k(N)]^T, \quad (52)$$

where T denotes the transpose operation. For updating network weights of CLDRNN following rule is used

$$W_{k+1} = W_k - (J_k^T J_k + \mu I)^{-1} g_k, \quad (53)$$

where W is the weight vector, J represents Jacobian matrix, μ denotes the scalar coefficient, which controls learning, I represents identity matrix, and the term g represents the gradient vector.

If the following terms are introduced in (53) as:

$$\Delta W_k = W_{k+1} - W_k, \quad (54)$$

and

$$\eta_k = (J_k^T J_k + \mu I)^{-1}, \quad (55)$$

where ΔW is the weight deviation vector and η represents learning rate. Now, substituting the expressions of (54) and (55) in (53), to get

$$\Delta W_k = -\eta_k g_k. \quad (56)$$

At k th epoch CLDRNN weight vector is defined as:

$$W_k = [w_k^1, w_k^2, \dots, w_k^M]^T, \quad (57)$$

where M represents weight vector length. For each network weight, the Jacobian and gradient are computed by:

$$J_k^j = \frac{\partial \mathcal{E}_k}{\partial w_k^j}, \quad (58)$$

and

$$g_k^j = \frac{1}{N} \sum_{n=1}^N \left(\varepsilon_k(n) \frac{\partial \varepsilon_k(n)}{\partial w_k^j} \right), \quad (59)$$

by using (55), (56) and (59), one can compute weight deviation in each weight as:

$$\Delta w_k = -\eta_k \frac{1}{N} \sum_{n=1}^N \left(\varepsilon_k(n) \frac{\partial \varepsilon_k(n)}{\partial w_k} \right). \quad (60)$$

In following discussion, a suitable choice of learning rate η is computed to speedup CLDRNN training. The concept of Lyapunov theory [23, 44, 66, 67] is adapted here to derive and prove the following *Theorem 3*.

Theorem 3 During CLDRNN training, at k th epoch, the learning rate η_k should met the following condition to reduce cost function value:

$$0 < \eta_k < \frac{2}{\frac{1}{N} \sum_{n=1}^N \left(\frac{\partial \hat{y}_k(n)}{\partial w_k} \right)^2}. \quad (61)$$

Proof To prove this, define a Lyapunov function as

$$V_k = N \times F_k = \frac{1}{2} \sum_{n=1}^N \varepsilon_k^2(n), \quad (62)$$

let the Lyapunov function deviation is defined as:

$$\Delta V_k = V_{k+1} - V_k = \frac{1}{2} \sum_{n=1}^N \left(\varepsilon_{k+1}^2(n) - \varepsilon_k^2(n) \right). \quad (63)$$

If the difference error deviation is defined as:

$$\Delta \varepsilon_k(n) = \frac{\partial \varepsilon_k(n)}{\partial w_k} \Delta w_k, \quad (64)$$

where ∂ is the partial derivative operator. Now by using (60), the expression in (64) can be rewritten as

$$\Delta \varepsilon_k(n) = -\eta_k \frac{\partial \varepsilon_k(n)}{\partial w_k} \frac{1}{N} \sum_{n=1}^N \left(\varepsilon_k(n) \frac{\partial \varepsilon_k(n)}{\partial w_k} \right), \quad (65)$$

since

$$\varepsilon_{k+1}(n) = \varepsilon_k(n) + \Delta \varepsilon_k(n), \quad (66)$$

by putting (65) and (66) into (63), to have

$$\Delta V_k = \frac{1}{2} \sum_{n=1}^N (\varepsilon_{k+1}(n) + \varepsilon_k(n)) (\varepsilon_{k+1}(n) - \varepsilon_k(n))$$

$$\Delta V_k = \frac{1}{2} \sum_{n=1}^N (2\varepsilon_k(n) + \Delta \varepsilon_k(n)) (\Delta \varepsilon_k(n)),$$

$$\Delta V_k = \sum_{n=1}^N \left(\varepsilon_k(n) \Delta \varepsilon_k(n) + \frac{1}{2} (\Delta \varepsilon_k(n))^2 \right)$$

$$\Delta V_k = \sum_{n=1}^N \left(\varepsilon_k(n) \frac{\partial \varepsilon_k(n)}{\partial w_k} \Delta w_k + \frac{1}{2} \left(\frac{\partial \varepsilon_k(n)}{\partial w_k} \Delta w_k \right)^2 \right) \quad (67)$$

by substituting (60) in (67), one can have

$$\begin{aligned} \Delta V_k &= -\eta_k \frac{1}{N} \left(\sum_{n=1}^N \varepsilon_k(n) \frac{\partial \varepsilon_k(n)}{\partial w_k} \right)^2 \\ &\quad + (\eta_k)^2 \frac{1}{2} \sum_{n=1}^N \left(\frac{\partial \varepsilon_k(n)}{\partial w_k} \right)^2 \left(\frac{1}{N} \sum_{n=1}^N \varepsilon_k(n) \frac{\partial \varepsilon_k(n)}{\partial w_k} \right)^2 \\ \Delta V_k &= \frac{\eta_k}{N} \left(\sum_{n=1}^N \varepsilon_k(n) \frac{\partial \varepsilon_k(n)}{\partial w_k} \right)^2 \left(\frac{\eta_k}{2N} \sum_{n=1}^N \left(\frac{\partial \varepsilon_k(n)}{\partial w_k} \right)^2 - 1 \right). \end{aligned} \tag{68}$$

Now, observe the three terms η_k , $\frac{1}{N} \sum_{n=1}^N \left(\frac{\partial \varepsilon_k(n)}{\partial w_k} \right)^2$ and $\left(\frac{\eta_k}{2N} \sum_{n=1}^N \left(\frac{\partial \varepsilon_k(n)}{\partial w_k} \right)^2 - 1 \right)$, in (68), where the learning rate in the first term should be $\eta_k > 0$. The second term will remain positive $\frac{1}{N} \sum_{n=1}^N \left(\frac{\partial \varepsilon_k(n)}{\partial w_k} \right)^2 \geq 0$.

Therefore, to satisfy the Lyapunov criterion to achieve $\Delta V_k < 0$, which is also equivalent to get

$$V_{k+1} < V_k \text{ or } F_{k+1} < F_k. \tag{69}$$

The third term in (68), must be negative as

$\left(\frac{\eta_k}{2N} \sum_{n=1}^N \left(\frac{\partial \varepsilon_k(n)}{\partial w_k} \right)^2 - 1 \right) < 0$. Hence, by considering these facts, and using the relation in (51), as $\frac{\partial \varepsilon_k(n)}{\partial w_k} = -\frac{\partial \hat{y}_k(n)}{\partial w_k}$, one can have the condition for adaptive learning rate as

$$0 < \eta_k < \frac{2}{\frac{1}{N} \sum_{n=1}^N \left(\frac{\partial \hat{y}_k(n)}{\partial w_k} \right)^2}. \tag{70}$$

Remark 5 Note that, the condition of *Theorem 3* on the adaptive learning rate η_k is not sufficient to achieve global convergence condition $-1 < \hat{\theta}_f < 1$. This is due to the multi-model nature of the cost function optimized for the identification of time-delayed systems. However, the global convergence can be obtained by utilizing an iterative approach presented in the proposed *Algorithm 1* with adaptive training of the CLDRNN until satisfying the global convergence condition $-1 < \hat{\theta}_f < 1$.

The complete proposed methodology is summarized as sequential steps in *Algorithm 1*. Moreover, a connection architecture is presented in Fig. 4, which shows the detailed interaction between the proposed algorithm and the CLDRNN to identify and model an unknown unstable delayed dynamics models.

Algorithm 1: Proposed Identification Algorithm using CLDRNN

Define:

- k_{max} : maximum epochs,
- μ : scalar coefficient
- μ_f : scalar multiplier
- μ_{max} : scalar’s maximum limit
- F^* : Cost function goal
- g_{min} : minimum gradient

1. Select a PID (K_p, K_i, K_d, β) controller to generate identification data;
2. Construct the CLDRNN architecture according to Figure 3;
3. Use PID controller gains (K_p, K_i, K_d, α) to set the weights in the CLDRNN’s controller layers using (3) to (7) and disable their learning;

for Iteration $i = 1$: converge **do**

(a) Use $\hat{\theta}_i^j = 0$, for the 1st iteration otherwise, use the value computed in step (c) to update CLDRNN’s architecture. Initialize μ and set the process model layer weights to ‘zero’.

for Epoch $k = 1$: k_{max} **do**

- I. Compute F_k, Ξ_k, J_k and g_k using (50), (52), (58) and (59);
- II. Compute η_k using (55), while validating *Theorem 3* in (61);
- III. **If** $\{(\mu = \mu_{max}) \text{ or } (F_k \leq F^*) \text{ or } (g_k \leq g_{min})\}$; **Then** break ;
- IV. Update W_{k+1} by (53) and compute F_{k+1} by (50);
- V. **If** $(F_{k+1} < F_k)$ **Then** Update: $\mu = \mu \times \mu_f$; **Else** Update: $\mu = \mu / \mu_f$ and go to step II;

end

(b) Compute $\hat{\theta}_f$ using (27) or by using (47);

(c) **If** Global convergence $(-1 < \hat{\theta}_f < 1)$ of (48) is met; **Then** break ;

Else Compute: $\hat{\theta}_i^{j+1} = \hat{\theta}_i^j + \hat{\theta}_f - \text{rem}(\hat{\theta}_f, 1)$ using (49);

end

4. Use the optimized CLDRNN weights to return the UFOD model using (11) or the USOD model using (34).

6 Simulation Study

In this section, some examples of unstable process dynamics are used for validating the proposed identification method. The proposed *Algorithm 1* is applied to the CLDRNN architecture, which is then trained with closed-loop identification data to model the unstable time-delayed systems.

The MSE (mean square of errors) criteria for closed-loop data has been adapted here for the identified models comparison as:

$$\text{MSE} = \frac{1}{2N} \sum_{n=1}^N (y(n) - \hat{y}(n))^2 \tag{71}$$

where $y(n)$ is the actual output, $\hat{y}(n)$ is the output of the identified model, and N is the number of training samples. To test the identification robustness of proposed method a white Gaussian noise as disturbance signal $d(n)$ is added in the true output. The mean value of noise signal is (m) considered ‘zero’, while variance (σ^2) is ‘nonzero’. The resulting signal

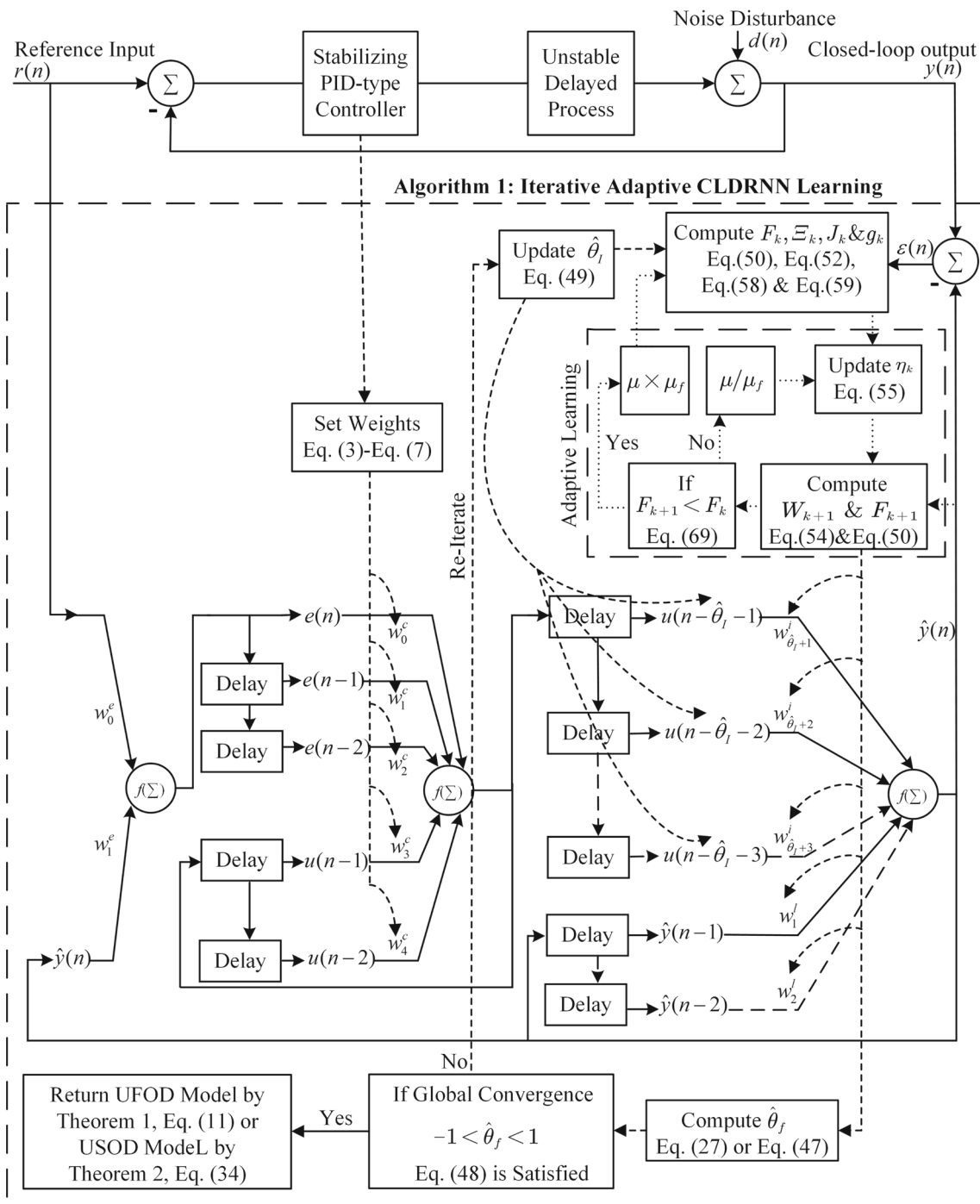


Fig. 4 Connection architecture for the closed-loop time-delayed process identification using CLDRNN

to noise ratio (SNR) is expressed in dB scale in terms of signal power (P_{signal}) and noise power (P_{noise}) as

$$\text{SNR} = 10 \log_{10} \left(\frac{P_{\text{signal}}}{P_{\text{noise}}} \right) \text{ dB.} \tag{72}$$

Some random experiments as Monte-Carlo tests are performed for analyzing the ability of the identification tech-

nique to handle the uncertainties caused due to measurement noise. Therefore, for a given SNR, total 100 Monte-Carlo tests where each simulation is performed with different noise seeds to test the accuracy and convergence of the proposed approach, where to represent the identification results each estimated parameter’s mean and standard deviation values are computed. All the simulation experiments of this section

Table 1 CLDRNN’s training parameters

Training parameters	Value
Maximum number of epochs (k_{\max})	50
Initial value of scalar coefficient (μ)	10^{-4}
Scalar multiplier (μ_f)	0.1
Scalar’s maximum limit (μ_{\max})	10^8
Cost function goal (F^*)	0.0
Minimum gradient (g_{\min})	10^{-6}

Table 2 Comparison of the identified model parameters for Example 1

Method	\hat{K}	$\hat{\tau}$	$\hat{\theta}$	MSE
Proposed	4.0001	-4.0002	2.0001	1.32×10^{-8}
Ananth et al. [45]	4.0110	-4.0690	2.0000	2.12×10^{-3}
Pandey et al. [57]	4.0018	-4.0033	2.0006	9.61×10^{-7}
Park et al. [50]	4.0100	-3.3250	2.0400	7.45×10^{-1}
Padhy et al. [16]	4.0051	-3.9997	2.0013	3.54×10^{-5}
True values	4.0000	-4.0000	2.0000	–

are performed using the MATLAB (Version R2018b) software.

6.1 Example 1

The first-order unstable process with time delay commonly used to model bio-reactors and chemical process plants. In this example, we consider an unstable first-order process, studied in [16, 45, 50, 57], as

$$G(s) = \frac{4}{4s - 1} e^{-2s}. \tag{73}$$

A PI controller with $K_p = 0.5$ and $K_i = 0.004$ is used to ensure the closed-loop stability of the system in (73). The closed-loop system is simulated with pseudo-random binary sequence (PRBS) input having 5000 samples each at 0.12 s interval and the clock period is 20 samples. This identification data is then used in the proposed CLDRNN identification Algorithm 1, with training parameters given in Table 1. The fixed controller layer weights computed by (3) to (7) as $w_0^c = 0.5$, $w_1^c = -0.9995$, $w_2^c = 0.4995$, $w_3^c = 2$, and $w_4^c = -1$. The identified model using the proposed approach compared with other literature methods in Table 2. All the identified models in Table 2 are validated using the MSE criterion in (71), for a closed-loop unit step response of 50 s duration. The identification results of Table 2 show that the model estimated by proposed approach produces minimum error compared to other methods.

The proposed method is also validated when the identification data is corrupted by the measurement noise with three

different cases of SNR’s of 20-dB, 30-dB, and 40-dB, respectively. A total of 100 Monte-Carlo simulations are performed with different noise seeds for each SNR value. The identification results presented in Table 3, where all the estimated process parameters are written in terms of their mean and standard deviation values. It is observed in Table 3, that the proposed method is robust and maintains identification consistency even in the presence of high measurement noise with accurate mean parameter and less standard deviation values.

Furthermore, in order to visualize the consistent identification accuracy of the proposed method, the case of 100 Monte-Carlo simulations with 20 dB SNR is considered. The results of all 100 simulations are plotted as the Nyquist graphs and the evolution of the CLDRNN weights with iterations in Figs. 5 and 6. The Nyquist plots in Fig. 5 show that all the identified model’s frequency responses accurately follow the true one. The evolution of the CLDRNN weights is shown in Fig. 6, where it can be observed that the weight corresponding to the unstable pole ($w_1^i > 1$) has small deviation and fast convergence, while the weights $w_{\hat{\theta}_l+1}^i$ and $w_{\hat{\theta}_l+2}^i$ has more deviation and takes more iterations to converge.

The evolution of the MSE cost function F_k with training epochs (k) is depicted in Fig. 7, where one can observe that for all the 100 Monte-Carlo simulations, the MSE (F_k) value decrease with each training epoch by using the adaptive learning rate as described in Theorem 3 and converges to a value nearer to zero.

To include the case of identifying time varying system a situation is considered, where it is assumed that the time delay of the system in (73) is changed from 1 s second to 2 s at time

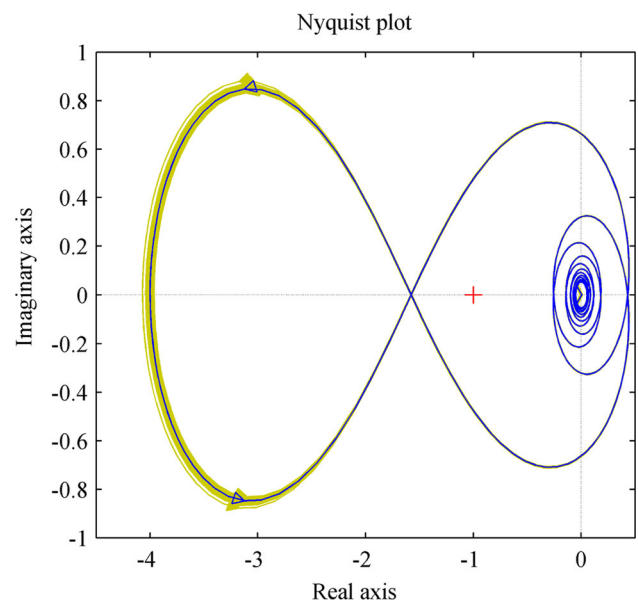


Fig. 5 Nyquist plots of the identified models for the 100 Monte-Carlo simulations of Example 1, with 20 dB SNR. Identified models (yellow line), actual process (blue line)

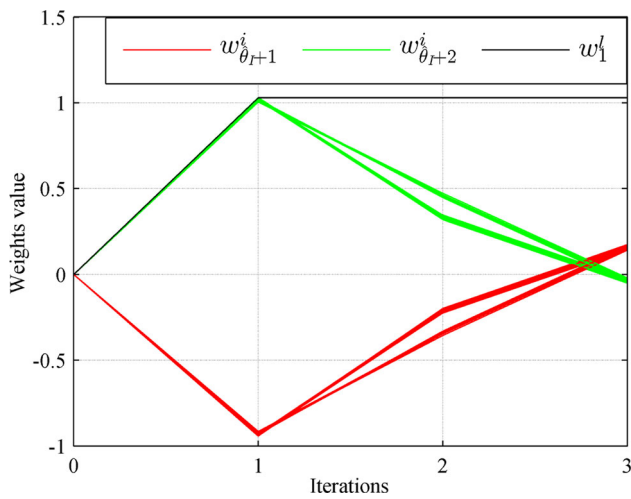


Fig. 6 Evolution of the CLDRNN’s process layer weights for the 100 Monte-Carlo simulations of Example 1, with 20 dB SNR

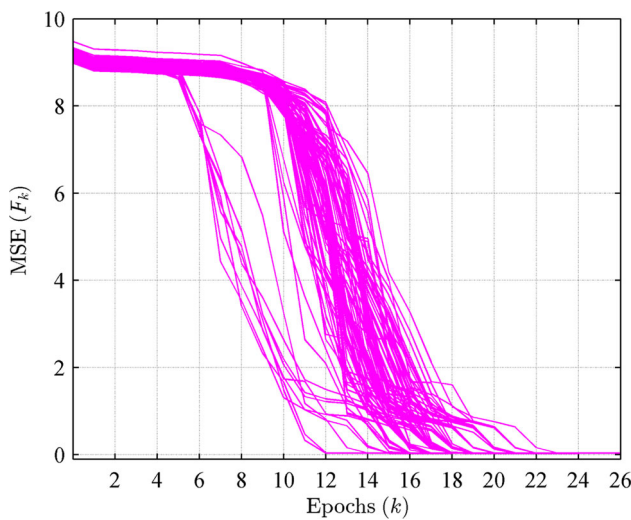


Fig. 7 Evolution and convergence of the MSE (F_k) for the 100 Monte-Carlo simulations of Example 1, with 20 dB SNR

1200 s. The identification experiment is performed for 2400 s with same settings. The proposed identification *Algorithm 1* is applied on batch data and the evolution of estimated parameters are shown in Fig. 8. As one can observe in Fig. 8 that the

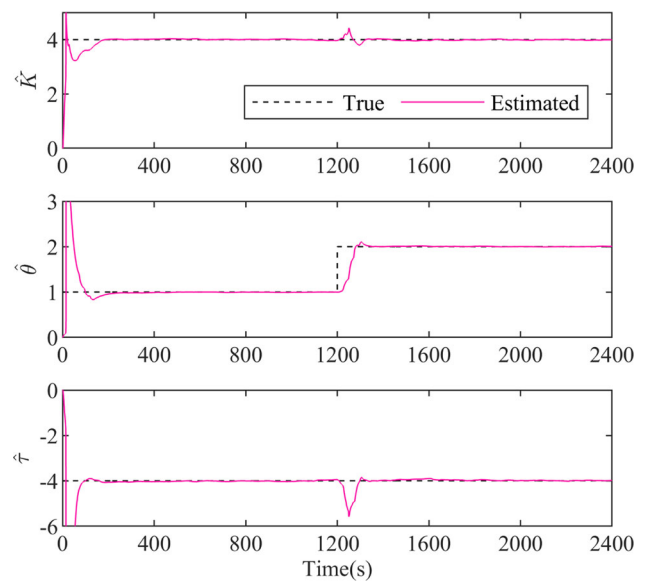


Fig. 8 Evolution of estimated parameters for varying time delay in Example 1, with 20 dB SNR

Table 4 Comparison of the identified UFOD and USOD models for Example 2

Model	Identified model	MSE
Example 2	$\frac{1.0e^{-0.5s}}{1.0s^2+1.5s-1}$	–
Proposed USOD	$\frac{1.0001e^{-0.5004s}}{1.0003s^2+1.5001s-1}$	1.67×10^{-7}
USOD of [52]	$\frac{1.0000e^{-0.5001s}}{1.0000s^2+1.5020s-1}$	9.15×10^{-7}
USOD of [54]	$\frac{1.0e^{-0.5s}}{1.0191s^2+1.5095s-1}$	8.72×10^{-5}
USOD of [48]	$\frac{0.9999e^{-0.4999s}}{1.0000s^2+1.5000s-1}$	2.23×10^{-7}
USOD of [55]	$\frac{1.0e^{-0.5s}}{1.3925s^2+1.4195s-1}$	3.56×10^{-2}
USOD of [56]	$\frac{1.0e^{-0.5s}}{1.3793s^2+1.4137s-1}$	3.46×10^{-2}
Proposed UFOD	$\frac{0.9768}{2.1735s-1} e^{-1.0074s}$	3.06×10^{-3}
UFOD of [47]	$\frac{0.9657}{2.4278s-1} e^{-1.0416s}$	2.05×10^{-2}
UFOD of [46]	$\frac{1.061}{2.545s-1} e^{-1.06s}$	1.34×10^{-2}
UFOD of [50]	$\frac{1.002}{2.347s-1} e^{-1.067s}$	5.43×10^{-3}
UFOD of [17]	$\frac{0.7534}{2.1642s-1} e^{-1.0412s}$	5.12×10^{-1}

Table 3 Results of 100 Monte-Carlo simulation for Example 1

SNR	Example 1 ($\frac{4.0}{4.0s-1} e^{-2.0s}$)		
	Identified parameters		
	Gain (\hat{K})	Time constant ($\hat{\tau}$)	Delay ($\hat{\theta}$)
20-dB	4.0049(±0.0175)	– 4.0091(±0.0255)	2.0027(±0.0051)
30-dB	4.0011(±0.0112)	– 4.0064(±0.0142)	2.0021(±0.0032)
40-dB	4.0008(±0.0061)	– 4.0063(±0.0078)	2.0021(±0.0017)
True Parameters	4.0000	– 4.0000	2.0000

Table 5 Identified model parameters' mean and standard deviation values for 100 Monte-Carlo simulation of Example 2, with different SNR

SNR	Example 2 $\left(\frac{1.0}{(2.0s-1)(0.5s+1)}e^{-0.5s}\right)$ Identified parameters			
	\hat{K}	$\hat{\tau}_1$	$\hat{\tau}_2$	$\hat{\theta}$
20 dB	1.0001(±0.0015)	-2.0007(±0.0041)	0.5005(±0.0051)	0.4995(±0.0076)
30 dB	1.0001(±0.0009)	-2.0004(±0.0025)	0.5003(±0.0032)	0.4999(±0.0045)
40 dB	1.0000(±0.0005)	-2.0002(±0.0013)	0.5001(±0.0016)	0.5001(±0.0024)
True	1.0000	2.0000	0.5000	0.5000

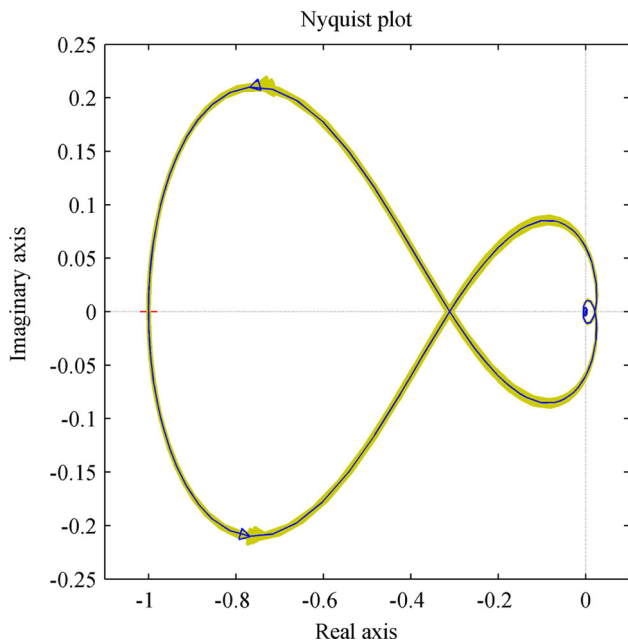


Fig. 9 Nyquist plots of the identified models for the 100 Monte-Carlo simulations of Example 2, with 20 dB SNR. Identified models (yellow line), actual system (blue line)

proposed method accurately tracks the changed time delay $\hat{\theta}$, without affecting the estimates of other model parameters.

6.2 Example 2

In general, the industrial food processing and biochemical units show the characteristics similar to a second-order unstable process with time delay. The following process model studied in literature by [46, 52, 55], used to test and validate the proposed technique as:

$$G(s) = \frac{1}{(2s-1)(0.5s+1)}e^{-0.5s}. \tag{74}$$

A PI controller with $K_p = 2$ and $K_i = 0.01$ is used to keep output bounded. A PRBS input is used having 5000 samples with 0.04 s sample time and clock period of 2.0 s for data generation. The proposed CLDRNN identification *Algorithm 1* is used with same training parameters as in Example

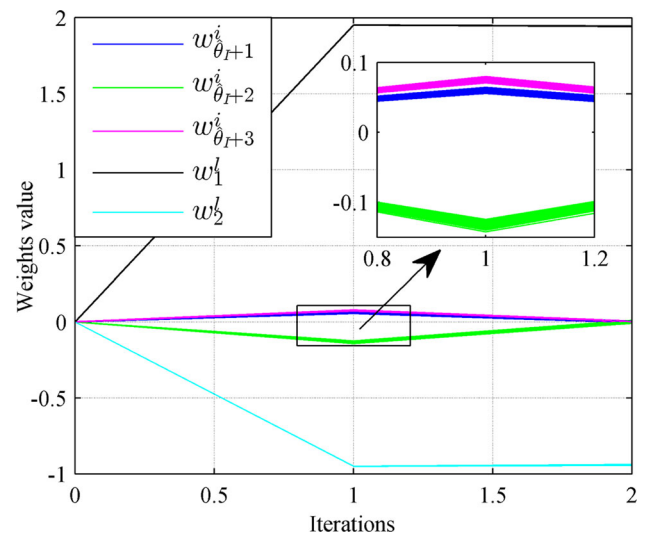


Fig. 10 Evolution of the CLDRNN's process layer weights for the 100 Monte-Carlo simulations of Example 2, with 20 dB SNR

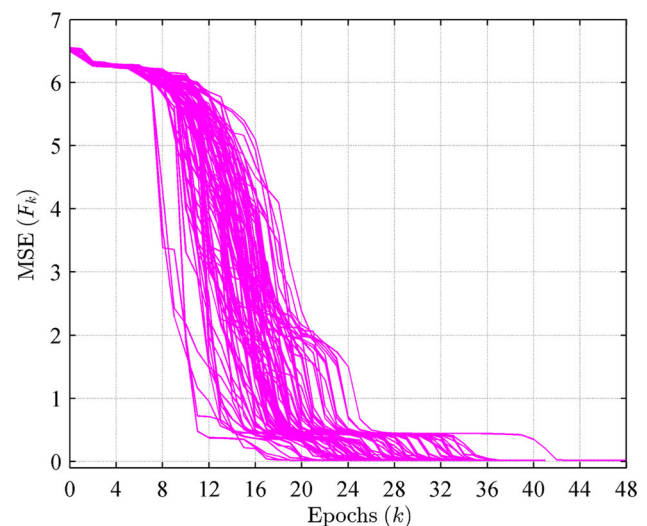


Fig. 11 Evolution and convergence of the MSE (F_k) for the 100 Monte-Carlo simulations of Example 2, with 20 dB SNR

1, having fixed controller layer weights computed by (3) to (7) as: $w_0^c = 2$, $w_1^c = -0.0026$, $w_2^c = 0.0069$, $w_3^c = 2$, and $w_4^c = -1$. In Table 4, the identified models are compared with other methods based on the MSE criterion described in (72). From the results of Table 4, one can say that the proposed approach is more accurate with lesser MSE values. In addition, for each noisy measurement of 20 dB, 30 dB, and 40 dB SNR, 100 random Monte-Carlo identification experiments are conducted. The simulation results are summarized in Table 5, where it is observed that the estimated parameters have accurate mean values with small deviations.

Moreover, for 20 dB SNR case, the 100 Monte-Carlo simulations are considered to visualize the consistent identification accuracy of the proposed method. The simulation results of all 100 identification experiments are depicted as the Nyquist graphs and evolution of network weights with iterations in Figs. 9 and 10, respectively. The Nyquist plots in Fig. 9 show that all the identified model's frequency responses accurately follow the true one. From the CLDRNN's weights evolution in Fig. 10, one can observe that the recurrent weights w_1^l and w_2^l have less deviation and fast convergence, while the weights $w_{\hat{\theta}_l+1}^i$, $w_{\hat{\theta}_l+2}^i$ and $w_{\hat{\theta}_l+3}^i$ have more deviation and takes more iterations to converge. The evolution of the MSE cost function F_k with training epochs (k) are depicted in Fig. 11, where one can observe that for all the 100 Monte-Carlo simulations, the MSE (F_k) value decrease with each training epoch by using the adaptive learning rate as described in Theorem 3 and converges at the proximity to zero.

6.3 Example 3

In this example, the proposed method is tested under the condition of model mismatch. A 3rd-order unstable delayed system studied by [58] is considered as:

$$G(s) = \frac{1.0e^{-0.5s}}{(5.0s - 1)(2.0s + 1)(0.5s + 1)} \quad (75)$$

A PID-type regulator having parameters $K_p = 1.5$, $K_i = 0.05$, $K_d = 0.6$ and $\beta = 10$, is used for the system in (75) to keep the output bounded. The closed-loop system is simulated to generate identification data using a PRBS input of 200 s duration with 0.08 s sampling interval and 4 s clock period. This identification data is then used in the proposed CLDRNN identification algorithm for system modeling with fixed controller layer weights computed using expressions (3) to (7) as $w_0^c = 7.5$, $w_1^c = 0.0043$, $w_2^c = -0.0077$, $w_3^c = 1.2$, and $w_4^c = -0.2$. The identified models are included for comparison and presented in Table 6. All the identified models in Table 6 are validated using MSE criterion in (72), for a closed loop unit step response of 50 s duration. The results shown

Table 6 Comparison of identified UFOD and USOD models for Example 3

Model	Identified model	MSE
Example 3	$\frac{1.0e^{-0.5s}}{(5.0s-1)(2.0s+1)(0.5s+1)}$	—
Proposed UFOD	$\frac{0.9757e^{-3.1356s}}{5.4691s-1}$	1.12×10^{-2}
Proposed USOD	$\frac{1.0002e^{-0.9765s}}{10.2665s^2+2.9684s-1}$	1.63×10^{-5}
UFOD model [53]	$\frac{1.0000e^{-1.9683s}}{5.7671s-1}$	3.28×10^{-1}
USOD model [58]	$\frac{1.001e^{-0.938s}}{10.354s^2+2.932s-1}$	2.58×10^{-5}

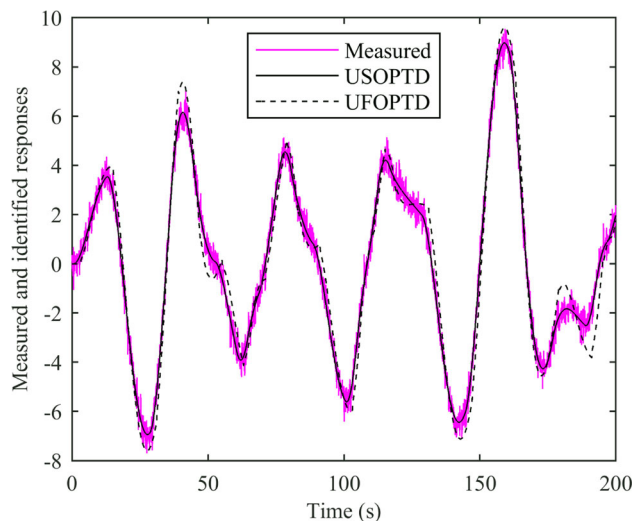


Fig. 12 Identified models responses in comparison to the measured response having 25 dB SNR for Example 3

in Table 6 indicate that the accuracy of proposed technique is better than other techniques.

In addition, the proposed algorithm is tested when the identification data is noisy with 20 dB SNR. The identified UFOD and USOD models for a case of noisy measurements are presented as follows:

$$\hat{G}_{\text{UFOD}}(s) = \frac{0.9759e^{-3.1378s}}{5.4719s - 1}, \quad (76)$$

and

$$\hat{G}_{\text{USOD}}(s) = \frac{1.0018e^{-0.9775s}}{10.3214s^2 + 2.9603s - 1}. \quad (77)$$

The measured output and the identified model's responses for a single simulation run are plotted in Fig. 12, which shows that both identified UFOD and USOD models follow the measured output, with the latter one being more accurate than the former one. The Nyquist plots of the identified model are depicted in Fig. 13, for a particular case of noisy measurements. In Fig. 13, it can be observed that the USOD model

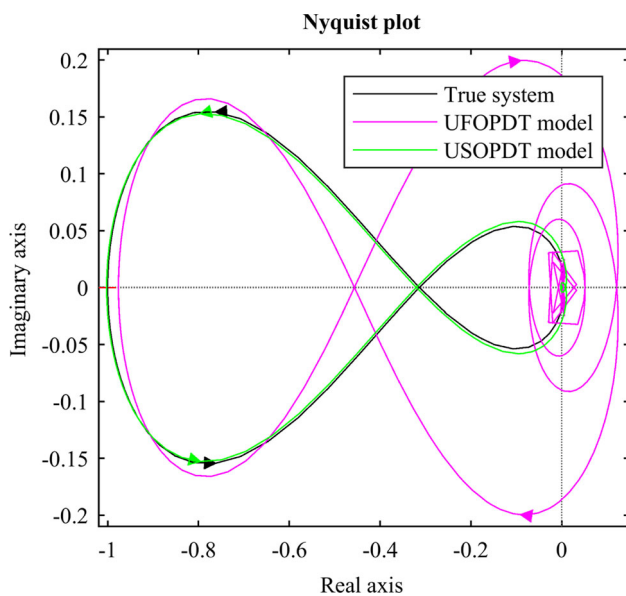


Fig. 13 Nyquist plots of identified models in comparison to the true system of Example 3

closely follows the actual system for all frequencies as compared to the UFOD model, which is accurate only nearer to the critical point of $(-1 \pm 0j)$ in the Nyquist plot.

7 Conclusion

For solving the problem of closed-loop identification of unstable delayed dynamical systems has been reported here. A neural architecture with delayed links is proposed which iteratively trains to ensure guaranteed convergence even with noisy measurements. The proposed CLDRNN architecture can mimic the complete closed-loop system within its architecture, including the controller information. The unknown unstable delayed system parameters can be directly extracted through CLDRNN. The benchmark unstable process model examples are incorporated for validating the algorithm, where the effects of measurement noise and model mismatch are also considered. The simulated experiments confirm that the proposed method is more general, accurate, robust and consistent as compared to the existing literature methods.

The directions for future work will be to extend the proposed concept to model time-delayed deep neural architectures to include further complex system dynamics with unknown time delays. Furthermore, the proposed idea could also be combined with the model reference adaptive control (MRAC) approach to carry out both identification and controller design simultaneously using the same neural network architecture. Also, the proposed concept will be more prominent in large-scale networked systems, where each sub-

system operating in closed-loop required to be identified and controlled.

References

1. He, R.; Chen, G.; Dong, C.; Sun, S.; Shen, X.: Data-driven digital twin technology for optimized control in process systems. *ISA Trans.* **95**, 221–234 (2019). <https://doi.org/10.1016/j.isatra.2019.05.011>
2. Zhang, Y.; Sun, L.; Shen, J.; Lee, K.Y.; Zhong, Q.C.: Iterative tuning of modified uncertainty and disturbance estimator for time-delay processes: a data-driven approach. *ISA Trans.* **84**, 164–177 (2019). <https://doi.org/10.1016/j.isatra.2018.08.028>
3. Lo, W.; Rad, A.; Li, C.: Self-tuning control of systems with unknown time delay via extended polynomial identification. *ISA Trans.* **42**(2), 259–272 (2003). [https://doi.org/10.1016/S0019-0578\(07\)60131-1](https://doi.org/10.1016/S0019-0578(07)60131-1)
4. Sung, S.W.; Lee, I.B.: Prediction error identification method for continuous-time processes with time delay. *Ind. Eng. Chem. Res.* **40**(24), 5743–5751 (2001). <https://doi.org/10.1021/ie0100636>
5. Kaya, I.; Atherton, D.: Parameter estimation from relay autotuning with asymmetric limit cycle data. *J. Process Control* **11**(4), 429–439 (2001). [https://doi.org/10.1016/S0959-1524\(99\)00073-6](https://doi.org/10.1016/S0959-1524(99)00073-6)
6. Alfaro, V.M.; Vilanova, R.: Optimal robust tuning for IDOF PI/PID control unifying FOPDT/SOPDT models. In: *IFAC Proceedings, 2nd IFAC Conference on Advances in PID Control*, vol. 45(3), 5pp. 72–577 (2012). <https://doi.org/10.3182/20120328-3-IT-3014.00097>
7. Anbarasan, K.; Srinivasan, K.: Design of RTDA controller for industrial process using SOPDT model with minimum or non-minimum zero. *ISA Trans.* **57**, 231–244 (2015). <https://doi.org/10.1016/j.isatra.2015.02.016>
8. Panda, R.C.; Yu, C.C.; Huang, H.P.: PID tuning rules for SOPDT systems: review and some new results. *ISA Trans.* **43**(2), 283–295 (2004). [https://doi.org/10.1016/S0019-0578\(07\)60037-8](https://doi.org/10.1016/S0019-0578(07)60037-8)
9. Kumar, A.; Saxena, S.: Parameter estimation in a system of integro-differential equations with time-delay. In: *IEEE Transactions on Circuits and Systems II: Express Briefs* pp 1–1 (2023). <https://doi.org/10.1109/TCSII.2023.3276080>
10. Bayrak, A.; Tatlicioglu, E.: A novel adaptive time delay identification technique. *ISA Trans.* (2023). <https://doi.org/10.1016/j.isatra.2023.05.001>
11. Bobal, V.; Kubalcik, M.; Dostal, P.; Matejcek, J.: Adaptive predictive control of time-delay systems. *Comput. Math. Appl.* **66**(2), 165–176 (2013). <https://doi.org/10.1016/j.camwa.2013.01.035>
12. Srivastava, S.; Pandit, V.: A PI/PID controller for time delay systems with desired closed loop time response and guaranteed gain and phase margins. *J. Process Control* **37**, 70–77 (2016). <https://doi.org/10.1016/j.jprocont.2015.11.001>
13. Wang, Y.J.: Determination of all feasible robust PID controllers for open-loop unstable plus time delay processes with gain margin and phase margin specifications. *ISA Trans.* **53**(2), 628–646 (2014). <https://doi.org/10.1016/j.isatra.2013.12.037>
14. Araujo, J.M.; Santos, T.L.: Control of second-order asymmetric systems with time delay: Smith predictor approach. In: *Mechanical Systems and Signal Processing*, vol. 137, p. 106355 (2020). Special issue on control of second-order vibrating systems with time delay. <https://doi.org/10.1016/j.ymsp.2019.106355>
15. Cong, E.D.; Hu, M.H.; Tu, S.T.; Xuan, F.Z.; Shao, H.H.: A novel double loop control model design for chemical unstable processes. *ISA Trans.* **53**(2), 497–507 (2014). <https://doi.org/10.1016/j.isatra.2013.11.003>



16. Padhy, P.; Majhi, S.: Relay based PI–PD design for stable and unstable FOPDT processes. *Comput. Chem. Eng.* **30**(5), 790–796 (2006). <https://doi.org/10.1016/j.compchemeng.2005.12.013>
17. Vivek, S.; Chidambaram, M.: An improved relay auto tuning of PID controllers for unstable FOPTD systems. *Comput. Chem. Eng.* **29**(10), 2060–2068 (2005). <https://doi.org/10.1016/j.compchemeng.2005.05.004>
18. Acharya, D.; Swain, S.; Mishra, S.: Real-time implementation of a stable 2 DOF PID controller for unstable second-order magnetic levitation system with time delay. *Arab. J. Sci. Eng.* **45**, 6311–6329 (2020). <https://doi.org/10.1007/s13369-020-04425-6>
19. Wang, S.; Yin, X.; Zhang, Y.; Li, P.; Wen, H.: Event-triggered cognitive control for networked control systems subject to dos attacks and time delay. *Arab. J. Sci. Eng.* **48**(5), 6991–7004 (2023). <https://doi.org/10.1007/s13369-022-07068-x>
20. Ha, H.; Welsh, J.S.; Alamir, M.: Useful redundancy in parameter and time delay estimation for continuous-time models. *Automatica* **95**, 455–462 (2018). <https://doi.org/10.1016/j.automatica.2018.06.023>
21. Forssell, U.; Ljung, L.: Identification of unstable systems using output error and Box–Jenkins model structures. *IEEE Trans. Autom. Control* **45**(1), 137–141 (2000). <https://doi.org/10.1109/9.827371>
22. Forssell, U.; Ljung, L.: Closed-loop identification revisited. *Automatica* **35**(7), 1215–1241 (1999). [https://doi.org/10.1016/S0005-1098\(99\)00022-9](https://doi.org/10.1016/S0005-1098(99)00022-9)
23. Cong, S.; Liang, Y.: PID-like neural network nonlinear adaptive control for uncertain multivariable motion control systems. *IEEE Trans. Ind. Electron.* **56**(10), 3872–3879 (2009). <https://doi.org/10.1109/TIE.2009.2018433>
24. Peng, J.; Dubay, R.: Nonlinear inversion-based control with adaptive neural network compensation for uncertain MIMO systems. *Expert Syst. Appl.* **39**(9), 8162–8171 (2012). <https://doi.org/10.1016/j.eswa.2012.01.151>
25. Li, X.; Zhang, W.; Fang, J.; Li, H.: Finite-time synchronization of memristive neural networks with discontinuous activation functions and mixed time-varying delays. *Neurocomputing* **340**, 99–109 (2019). <https://doi.org/10.1016/j.neucom.2019.02.051>
26. Li, X.; Fang, J.; Li, H.: Exponential adaptive synchronization of stochastic memristive chaotic recurrent neural networks with time-varying delays. *Neurocomputing* **267**, 396–405 (2017). <https://doi.org/10.1016/j.neucom.2017.06.049>
27. Zhang, X.M.; Han, Q.L.; Ge, X.; Ding, D.: An overview of recent developments in Lyapunov–Krasovskii functionals and stability criteria for recurrent neural networks with time-varying delays. *Neurocomputing* **313**, 392–401 (2018). <https://doi.org/10.1016/j.neucom.2018.06.038>
28. Arslan, E.: Novel criteria for global robust stability of dynamical neural networks with multiple time delays. *Neural Netw.* **142**, 119–127 (2021). <https://doi.org/10.1016/j.neunet.2021.04.039>
29. Li, J.; Wang, Z.; Dong, H.; Ghinea, G.: Outlier-resistant remote state estimation for recurrent neural networks with mixed time-delays. *IEEE Trans. Neural Netw. Learn. Syst.* **32**(5), 2266–2273 (2021). <https://doi.org/10.1109/TNNLS.2020.2991151>
30. Mainer, D.; Mansour, M.; Hamache, A.; Boudjedir, C.; Bounabi, M.: Switched time delay control based on artificial neural network for fault detection and compensation in robot manipulators. *SN Appl. Sci.* **3**, 1–13 (2021). <https://doi.org/10.1007/s42452-021-04376-z>
31. Tutunji, T.A.: Approximating transfer functions using neural network weights. In: 2009 4th International IEEE/EMBS Conference on Neural Engineering, pp. 641–644 (2009). <https://doi.org/10.1109/NER.2009.5109378>
32. Sharma, S.; Padhy, P.K.: Discrete transfer function modeling of non-linear systems using neural networks. In: 2019 Fifth International Conference on Image Information Processing (ICIIP), pp 558–563 (2019). <https://doi.org/10.1109/ICIIP47207.2019.8985827>
33. Mutasa, S.; Sun, S.; Ha, R.: Understanding artificial intelligence based radiology studies: what is overfitting? *Clin. Imaging* **65**, 96–99 (2020). <https://doi.org/10.1016/j.clinimag.2020.04.025>
34. Tangirala, A.: Principles of System Identification: Theory and Practice. CRC Press, Boca Raton (2014)
35. Trischler, A.P.; D’Eleuterio, G.M.: Synthesis of recurrent neural networks for dynamical system simulation. *Neural Netw.* **80**, 67–78 (2016). <https://doi.org/10.1016/j.neunet.2016.04.001>
36. Kumar, R.; Srivastava, S.; Gupta, J.; Mohindru, A.: Temporally local recurrent radial basis function network for modeling and adaptive control of nonlinear systems. *ISA Trans.* **87**, 88–115 (2019). <https://doi.org/10.1016/j.isatra.2018.11.027>
37. Narendra, K.S.; Parthasarathy, K.: Identification and control of dynamical systems using neural networks. *IEEE Trans. Neural Netw.* **1**(1), 4–27 (1990). <https://doi.org/10.1109/72.80202>
38. Ho, H.; Rad, A.; Wong, Y.; Lo, W.: On-line lower-order modeling via neural networks. *ISA Trans.* **42**(4), 577–593 (2003). [https://doi.org/10.1016/S0019-0578\(07\)60007-X](https://doi.org/10.1016/S0019-0578(07)60007-X)
39. Sharma, S.; Verma, B.; Trivedi, R.; Padhy, P.K.: Identification of stable FOPDT process parameters using neural networks. In: 2018 International Conference on Power Energy, Environment and Intelligent Control (PEEIC), pp. 545–549 (2018). <https://doi.org/10.1109/PEEIC.2018.8665411>
40. Tutunji, T.A.: Parametric system identification using neural networks. *Appl. Soft Comput.* **47**, 251–261 (2016). <https://doi.org/10.1016/j.asoc.2016.05.012>
41. Tutunji, T.A.; Saleem, A.: Weighted parametric model identification of induction motors with variable loads using FNN structure and nn2tf algorithm. *Trans. Inst. Meas. Control* **40**(5), 1645–1658 (2018). <https://doi.org/10.1177/0142331216688249>
42. Chon, K.H.; Cohen, R.J.: Linear and nonlinear ARMA model parameter estimation using an artificial neural network. *IEEE Trans. Biomed. Eng.* **44**(3), 168–174 (1997). <https://doi.org/10.1109/10.554763>
43. Fei, M.; Zhang, J.; Hu, H.; Yang, T.: A novel linear recurrent neural network for multivariable system identification. *Trans. Inst. Meas. Control* **28**(3), 229–242 (2006). <https://doi.org/10.1191/0142331206tim171oa>
44. Peng, J.; Dubay, R.: Identification and adaptive neural network control of a dc motor system with dead-zone characteristics. *ISA Trans.* **50**(4), 588–598 (2011). <https://doi.org/10.1016/j.isatra.2011.06.005>
45. Ananth, I.; Chidambaram, M.: Closed-loop identification of transfer function model for unstable systems. *J. Frankl. Inst.* **336**(7), 1055–1061 (1999). [https://doi.org/10.1016/S0016-0032\(99\)00031-9](https://doi.org/10.1016/S0016-0032(99)00031-9)
46. Cheres, E.: Parameter estimation of an unstable system with a PID controller in a closed loop configuration. *J. Frankl. Inst.* **343**(2), 204–209 (2006). <https://doi.org/10.1016/j.jfranklin.2005.09.007>
47. Sree, R.P.; Chidambaram, M.: Improved closed loop identification of transfer function model for unstable systems. *J. Frankl. Inst.* **343**(2), 152–160 (2006). <https://doi.org/10.1016/j.jfranklin.2005.10.001>
48. Liu, T.; Gao, F.: Identification of low-order unstable process model from closed-loop step test. In: IFAC Proceedings, 7th IFAC Symposium on Advanced Control of Chemical Processes, vol. 42(11), pp. 447–451 (2009). <https://doi.org/10.3182/20090712-4-TR-2008.00071>
49. Herrera, J.; Ibeas, A.; Alcántara, S.; de la Sen, M.; Serna-Garcés, S.: Identification and control of delayed SISO systems through pattern search methods. *J. Frankl. Inst.* **350**(10), 3128–3148 (2013). <https://doi.org/10.1016/j.jfranklin.2013.06.022>

50. Park, J.H.; Sung, S.W.; Lee, I.B.: DAn enhanced PID control strategy for unstable processes. *Automatica* **34**(6), 751–756 (1998). [https://doi.org/10.1016/S0005-1098\(97\)00235-5](https://doi.org/10.1016/S0005-1098(97)00235-5)
51. Vivek, S.; Chidambaram, M.: Identification using single symmetrical relay feedback test. *Comput. Chem. Eng.* **29**(7), 1625–1630 (2005). <https://doi.org/10.1016/j.compchemeng.2005.01.002>
52. Pandey, S.; Majhi, S.: Relay-based identification scheme for processes with non-minimum phase and time delay. *IET Control Theory Appl.* **13**(15), 2507–2519 (2019). <https://doi.org/10.1049/iet-cta.2018.6170>
53. Marchetti, G.; Scali, C.; Lewin, D.: Identification and control of open-loop unstable processes by relay methods. *Automatica* **37**(12), 2049–2055 (2001). [https://doi.org/10.1016/S0005-1098\(01\)00181-9](https://doi.org/10.1016/S0005-1098(01)00181-9)
54. Bajarangbali, R.; Majhi, S.; Pandey, S.: Identification of FOPDT and SOPDT process dynamics using closed loop test. *ISA Trans.* **53**(4), 1223–1231 (2014). <https://doi.org/10.1016/j.isatra.2014.05.014>
55. Bajarangbali, R.; Majhi, S.: Estimation of first and second order process model parameters. *Proc. Natl. Acad. Sci. India Sect. A* **88**(4), 557–563 (2018). <https://doi.org/10.1007/s40010-017-0357-6>
56. Bajarangbali, M.S.: Modeling of stable and unstable second order systems with time delay. In: 2013 Annual IEEE India Conference (INDICON), pp 1–5 (2013). <https://doi.org/10.1109/INDICON.2013.6725904>
57. Pandey, S.; Majhi, S.: Identification and control of unstable FOPTD processes with improved transients. *Electron. Lett.* **53**(5), 312–314 (2017). <https://doi.org/10.1049/el.2016.3769>
58. Majhi, S.: Relay based identification of processes with time delay. *J. Process Control* **17**(2), 93–101 (2007). <https://doi.org/10.1016/j.jprocont.2006.09.005>
59. Agüero, J.C.; Goodwin, G.C.; Van den Hof, P.M.: A virtual closed loop method for closed loop identification. *Automatica* **47**(8), 1626–1637 (2011). <https://doi.org/10.1016/j.automatica.2011.04.014>
60. Maruta, I.; Sugie, T.: Stabilized prediction error method for closed-loop identification of unstable systems. In: IFAC-PapersOnLine, 18th IFAC Symposium on System Identification SYSID 2018, vol. 51(15), pp. 479–484 (2018). <https://doi.org/10.1016/j.ifacol.2018.09.191>
61. Aljanaideh, K.; Coffey, B.J.; Bernstein, D.S.: Closed-loop identification of unstable systems using noncausal fir models. In: 2013 American Control Conference, pp 1669–1674 (2013). <https://doi.org/10.1109/ACC.2013.6580075>
62. Ljung, L.: *System Identification: Theory for the User*. Prentice Hall Information and System Sciences Series. Prentice Hall PTR, Hoboken (1999)
63. Karimi, A.; Landau, I.D.: Comparison of the closed-loop identification methods in terms of the bias distribution. *Syst. Control Lett.* **34**(4), 159–167 (1998). [https://doi.org/10.1016/S0167-6911\(97\)00137-0](https://doi.org/10.1016/S0167-6911(97)00137-0)
64. Du, Y.Y.; Tsai, J.S.; Patil, H.; Shieh, L.S.; Chen, Y.: Indirect identification of continuous-time delay systems from step responses. *Appl. Math. Model.* **35**(2), 594–611 (2011). <https://doi.org/10.1016/j.apm.2010.07.004>
65. Sharma, S.; Padhy, P.K.: A novel iterative system identification and modeling scheme with simultaneous time-delay and rational parameter estimation. *IEEE Access* **8**, 64918–64931 (2020). <https://doi.org/10.1109/ACCESS.2020.2985132>
66. Kumar, R.; Srivastava, S.; Gupta, J.; Mohindru, A.: Diagonal recurrent neural network based identification of nonlinear dynamical systems with Lyapunov stability based adaptive learning rates. *Neurocomputing* **287**, 102–117 (2018). <https://doi.org/10.1016/j.neucom.2018.01.073>
67. Kumar, R.; Srivastava, S.: Externally recurrent neural network based identification of dynamic systems using Lyapunov stability analysis. *ISA Trans.* **98**, 292–308 (2020). <https://doi.org/10.1016/j.isatra.2019.08.032>

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.

