AUT Journal of Modeling and Simulation

AUT J. Model. Simul., 51(2) (2019) 199-210
DOI: 10.22060/miscj.2019.16029.5151

Iterative learning identification and control for dynamic systems described by NARMAX model

F. Afsharnia[1], A. Madady[1,*], M. B. Menhaj[2]

[1] Department of Electrical Engineering, Tafresh University, Tafresh, Iran
[2] Department of Electrical Engineering, Amirkabir University of Technology, Tehran, Iran

ABSTRACT: A new iterative learning controller is proposed for a general unknown discrete time-varying nonlinear non-affine system represented by NARMAX (Nonlinear Autoregressive Moving Average with eXogenous inputs) model. The proposed controller is composed of an iterative learning neural identifier and an iterative learning controller. Iterative learning control and iterative learning identification are integrated in each iteration. A multi-layer neural network is used for identification. Since the system considered in this paper is time-varying, the proposed neural identifier also is time-varying. The weights of the neural identifier are updated at each iteration, so both tracking performance and identification are improved at each iteration simultaneously. The structure of the proposed neural network used for identification system is affine in control input. Then new iterative learning control law based on the neural identifier is proposed and applied to the system. It should be mentioned that the proposed integrated algorithm has a faster, better and more accurate performance when compared with other iterative learning control algorithms proposed for similar systems. Convergence of both the trajectory tracking error and identification error is guaranteed along the iteration domain with repeating the process within a time-limited range. Simulation and comparison results easily approve the effectiveness of the proposed method.

Review History:

Received: 2019-03-19
Revised: 2019-06-17
Accepted: 2019-06-19
Available Online: 2019-12-01

Keywords:

iterative learning control
iterative learning identification
NARMAX model
neural network

1. Introduction

Iterative learning control for robotic manipulators is initially proposed by Arimoto et al. [1], then has been widely developed in theoretical and applied fields such as robotics [2, 3], actuators (electromechanical valves) [4], batch processes [5], RF power amplifier linearization [6] and etc. It is a useful control method for improving transient response and tracking performance of the reference signal for unknown or uncertain dynamical systems that repeat a task in a finite time interval. In various control environment, learning methods are a complementary approach to the existing robust and adaptive control techniques. Robust and adaptive control techniques ensure the convergence along the time axis while learning control can guarantee the convergence along the iteration domain by repeating the process within a time-limited range [7-15].

Iterative learning identification (ILI) is a new approach. ILI uses data from each iteration to estimate of the system parameters [16], It is particularly effective for linear time-varying systems that their parameters vary quickly [17]. For linear time-varying and discrete time systems described by the autoregressive model (ARX), an ILC approach and an ILI are combined in [18], the ILI technique used in [18] is based on norm optimal.

Nonlinear Autoregressive Moving Average with eXogenous (NARMAX) model is a general description of discrete-time nonlinear non-affine systems. The NARMAX model was proposed in 1981 as a new description [19-21], then has been widely developed for A great variety of systems, such as space weather system, neuroimaging, finance, the solar-terrestrial, and stem cells dynamics. NARMAX model is one of the most robust techniques for considering complex dynamical-systems [22-25]. It is very difficult to design a controller for such a model.

In this paper, we consider both of iterative learning identification and control for a general class of unknown nonlinear systems. Since a multilayer feedforward network with as few as one hidden layer can approximate any nonlinear mapping to any desired degree of accuracy, artificial neural network is a effective tool to use in control processes, it can also learn and adapt to dynamical property of unknown system, so neural network based control system is mainly robust and adaptive [26-35].

A neural network based ILC for robot presented in [36], for tracking of the desired trajectory a compound control law that composed of the feed-forward ILC law and feedback law is applied to the system based on the model that identified by

*Corresponding author's email: drmadady@yahoo.com

the neural network. The neural network is trained to reach optimal weights in any iteration of the control algorithm, updating of neural network weights is also an iterative algorithm (the control iteration and identification process iteration are different) so the proposed approach has very much computing that takes a lot of time.

Based on an equivalent compact form dynamic linearization (CFDL) data model of the nonlinear system in the iteration domain, a model-free adaptive control (MFAC) for a class of discrete-time nonlinear systems has been presented in [37]. The proposed control design in [37] is not accurate. The neural controller has been used in [38, 39] to follow the reference trajectory, the neural controller is a multilayer neural network whose parameters are updated after each iteration. The training process of this neural controller needs an precise estimation of the system that be achieved with the use of another neural network modeling the controlled plant.

In all the reasearches mentioned before, there are defects that we have rectified them. In this paper, we propose a new ILC based on a neural network for a general unknown discrete time-varying nonlinear non-affine system described by NARMAX. The proposed approach in this paper is considered for a general class of nonlinear systems, whereas the above references [36-39] are applied to limited classes of systems. The proposed approach in this paper is a design of the integrated identification and control algorithm simultaneously, so our method takes less time than proposed methods in [36-39]. Also, this proposed approach has a better and more accurate performance than reasearches mentioned before.

A multilayer perceptron neural network (MLPNN) is used as an iterative identifier whose weights is updated along the control iteration domain. Since the mapping between the input and output of a time-varying system changes over time, the mapping between the input and output of the neural network must also be able to change over time [40], so the proposed neural network is time-varying. Descendant gradient algorithm is used for training of the weights of the neural network. The proposed ILC based on MLPNN is applied to the system such as the output of the neural network tracks the desired output trajectory, while the output of the neural network tends to the output of the system as iteration number increases (this approach is displayed in Fig. 1). The neural network output is rapidly converted to the given desired output trajectory but to converge the system output to the desired output trajectory must firstly identification be reached accurately. Convergence of both the trajectory tracking error and identification error is guaranteed along the iteration domain with repeating the process within a time-limited range

The paper is organized as follows. Section 2 defines the problem, the system, and the preliminaries. Section 3 presents the structure of the neural network used for identification of the system defined in the previous section, and its convergence analysis is presented . An ILC based on neural network is presented in Section 4. Section 5 gives the simulation example. The conclusion is given in section 6.

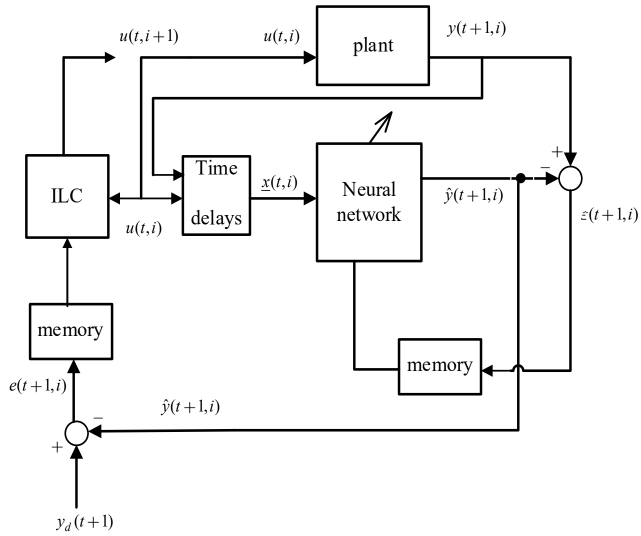

Fig. 1. General structure of iterative learning control based on neural networks

2. PROBLEM FORMULATION AND PRELIMINARIES

The underlying dynamic system to be identified and controlled is a general unknown and repeatable discrete time-varying nonlinear non-affine system described by NARMAX which is displayed as follows:

$$
\begin{aligned}
y(t+1,i) &= F\big[\, y(t,i), y(t-1,i),\dots, y(t-n_y,i),\\
&\quad u(t,i), u(t-1,i),\dots, u(t-n_u,i), t \,\big] \\
t &= 0,1,\dots,N \qquad , \qquad i = 0,1,..
\end{aligned}
\tag{1}
$$

where $u(t,i)\in R$ and $y(t,i)\in R$ are the controlled (i.e. exogenous) input and the output of the system, respectively. The index $i=0,1,\dots$ is the iteration or trial number, and $t\in[0,N]$ is the finite time interval a given trial, N denotes the time duration of the repetitions which is assumed to be fixed and known. n_y and n_u are the maximum lags for the system output, input, and error, respectively. $F[.]$ is an unknown nonlinear mapping.

Consider (1) and make the following reasonable assumptions:

A1) the system (1) is a relaxed system so

$$
\begin{aligned}
& y(0,i) = y(-1,i) = \dots = y(-n_y,i) = \\
& u(-1,i) = u(-2,i) = \dots = y(-n_u,i) = 0 \\
& \qquad\qquad i=0,1,\dots
\end{aligned}
$$

3. Iterative learning identification

A multilayer perceptron neural network is used as an iterative learning identifier. Since the mapping between the input and output of a time-varying system changes over time, the mapping between the input and output of the neural network must also be able to change over time, so the proposed neural network is time-varying [40]. Also, we

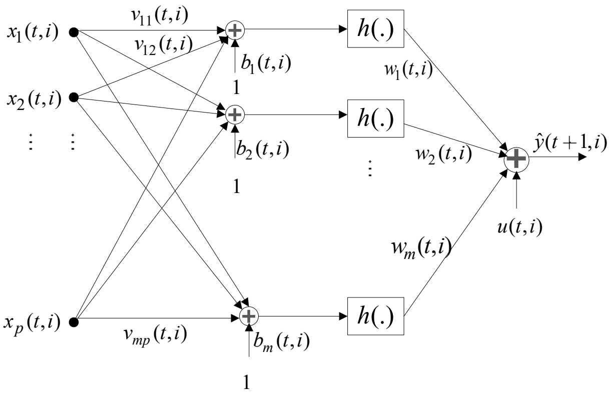

Fig. 2. The structure of the proposed MLPNN

propose the structure of MLPNN in such a way that it is affine in input $u(t,i)$ (see Fig. 1). The proposed MLPNN is shown in matrix form as follows:

$$\hat{y}(t+1,i) = \underline{w}(t,i)^T \, \underline{h}\begin{pmatrix} V(t,i)\underline{x}(t,i) + \\ \underline{b}(t,i) \end{pmatrix} + u(t,i) \tag{2}$$
$$t = 0,1,\ldots,N \ , \ i = 0,1,\ldots$$

$$\underline{x}(t,i) = \begin{bmatrix} y(t,i) & y(t-1,i) & \ldots & y(t-n_y,i) \\ u(t-1,i) & \ldots & u(t-n_u,i) \end{bmatrix}^T \tag{3}$$

Where $\underline{w}(t,i) \in R^{m\times 1}$ and $V(t,i) \in R^{m\times p}$ are the output-hidden weight matrix and hidden-input weight matrix, respectively, in which $p = n_u + n_y + 1$ and m is the number of hidden neurons. $\underline{x}(t,i) \in R^{p\times 1}$ and $u(t,i)$ are inputs of the neural network. The term $V(t,i)\underline{x}(t,i) + \underline{b}(t,i)$ can be rewritten as

$\underbrace{[V(t,i) \quad \underline{b}(t,i)]}_{V_a(t,i)} \underbrace{\begin{bmatrix} \underline{x}(t,i) \\ 1 \end{bmatrix}}_{\underline{x}_a(t,i)}$, so form (3) rewritten as:

$$\hat{y}(t+1,i) = \underline{w}(t,i)^T \, \underline{h}\left(V_a(t,i)\underline{x}_a(t,i)\right) + u(t,i) \tag{4}$$

Where $\underline{x}_a(t,i) = [\underline{x}(t,i)^T, 1]^T \in R^{(p+1)\times 1}$ and $V_a(t,i) \in R^{m\times(p+1)}$ are the augmented neural input vector (the –1 term signifies the input bias) and the augmented output-hidden weight matrix.

The vector $.\underline{h}\left(V_a(t,i)\underline{x}(t,i)\right)$ is defined as:

$$\underline{h}\left(V_a(t,i)\underline{x}(t,i)\right) = \begin{bmatrix} h\left(V_{a1}(t,i)\underline{x}(t,i)\right) \\ \vdots \\ h\left(V_{am}(t,i)\underline{x}(t,i)\right) \end{bmatrix}$$

Where $p = n_u + n_y + 1$ and m is the number of hidden neurons. $h(.)$ is a sigmoid function.

We define identification error $\varepsilon(t,i)$ and cost function as:

$$\underset{t=0,1,\ldots,N \quad i=0,1,\ldots}{\varepsilon(t+1,i) = y(t+1,i) - \hat{y}(t+1,i)} \tag{5}$$

$$J(t,i) = \frac{1}{2}\left(\varepsilon(t+1,i)\right)^2 \quad t=0,1,\ldots,N \quad i=0,1,\ldots \tag{6}$$

Using gradient descendant algorithm [41], the parameters of the neural network are updating with the following learning laws as:

$$V_a(t,i+1) = V_a(t,i) - \alpha_V(i)\left(\frac{\partial J(t,i)}{\partial V_a(t,i)}\right), \quad t=0,1,\ldots,N \quad i=0,1,\ldots \tag{7}$$

$$\underline{w}(t,i+1) = \underline{w}(t,i) - \alpha_w(i)\left(\frac{\partial J(t,i)}{\partial \underline{w}(t,i)}\right). \quad t=0,1,\ldots,N \quad i=0,1,\ldots \tag{8}$$

Where $\alpha_w(i)$ and $\alpha_V(i)$ are learning gains of the identifier neural network.

$$\frac{\partial J(t,i)}{\partial \underline{w}(t,i)} = \varepsilon(t+1,i)\frac{\partial \varepsilon(t+1,i)}{\partial \underline{w}(t,i)} \tag{9}$$

$$\frac{J(t,i)}{V_a(t,i)} = \varepsilon(t+1,i)\frac{\partial \varepsilon(t+1,i)}{\partial V_a(t,i)} \tag{10}$$

$$\frac{\partial \varepsilon(t+1,i)}{\partial \underline{w}(t,i)} = -\frac{\partial \hat{y}(t+1,i)}{\partial \underline{w}(t,i)} \tag{11}$$

$$\frac{\partial \varepsilon(t+1,i)}{\partial V_a(t,i)} = -\frac{\partial \hat{y}(t+1,i)}{\partial V_a(t,i)} \tag{12}$$

We define $\Delta\underline{w}^T(t,i) = \underline{w}(t,i+1) - \underline{w}(t,i)$ and $\Delta V_a(t,i) = V_a(t,i+1) - V_a(t,i)$. From equations (7)-(10), we have:

$$\Delta\underline{w}^T(t,i) = \alpha_w(i)\varepsilon(t+1,i)\frac{\partial \hat{y}(t+1,i)}{\partial \underline{w}(t,i)} \tag{13}$$

$$\Delta V_a(t,i) = \alpha_V(i)\varepsilon(t+1,i)\frac{\partial \hat{y}(t+1,i)}{\partial V_a(t,i)} \tag{14}$$

$$\frac{\partial \hat{y}(t+1,i)}{\partial \underline{w}(t,i)} = \underline{h}\left(V_a^T(t,i)\underline{x}_a(t,i)\right) \tag{15}$$

$$\frac{\partial \hat{y}(t+1,i)}{\partial V_a(t,i)} = \frac{\partial}{\partial V_a(t,i)}\left[\underline{w}(t,i)^T \underline{h}\left(V_a^T(t,i)\underline{x}_a(t,i)\right)\right] \tag{16}$$

$$\frac{\partial \hat{y}(t+1,i)}{\partial V_a(t,i)} = H'\left(V_a(t,i)\underline{x}_a(t,i)\right)\underline{w}(t,i)\underline{x}_a^T(t,i) \tag{17}$$

Remark 1. $H'\left(V_a(t,i)\underline{x}_a(t,i)\right)\in R^{m\times m}$ is a diagonal matrix with elements on the main diagonal $h'\left(\underline{v}_{al}^T(t,i)\underline{x}_a(t,i)\right)$ which in $\underline{v}_{al}^T(.)$ is lth raw of matrix $V_a(t,i)$ for $l=1,...,m$.

Theorem 1. Consider system (1), The proposed multilayer neural network (4) is proposed as an iterative learning identifier for (1). Using adaptive iterative updating laws (7) and (8), the convergence of the proposed neural network is guaranteed if learning gains are designed as follows:

$$0<\alpha_w(i)<\frac{1}{(\mu_w(i))^2} \tag{18}$$

$$0<\alpha_V(i)<\frac{1}{(\mu_V(i))^2} \tag{19}$$

$\mu_w(i)$ and $\mu_V(i)$ are defined as:

$$\mu_w(i) \triangleq \max_t \left\|\frac{\partial \hat{y}(t+1,i)}{\partial \underline{w}(t,i)}\right\| = \max_t \left\|\underline{h}\left(V_a^T(t,i)\underline{x}_a(t,i)\right)\right\| \tag{20}$$

$$\mu_V(i) \triangleq \max_t \left\|\frac{\partial \hat{y}(t+1,i)}{\partial \hat{V}_a(t,i)}\right\|_f = \max_t \left\|H'\left(V_a(t,i)\underline{x}_a(t,i)\right)\underline{w}(t,i)\underline{x}_a^T(t,i)\right\|_f \tag{21}$$

Where $\|.\|$ is the Euclidean norm and $\|.\|_f$ is the Frobenius norm. If a and A are a vector and a matrix:

$$\begin{cases}\|a\|=\sqrt{a^T a}\\ \|A\|_f=\sqrt{tr(A^T A)}\end{cases}$$

Proof: We define a Lyapunov's function as:

$$\Phi(t,i)=\frac{1}{2}\varepsilon^2(t+1,i) \quad {\scriptstyle t=0,1,...,N\ ,\ i=0,1,...} \tag{22}$$

The difference between Lyapunov functions of two consecutive iterations is defined as :

$$\Delta\Phi(t,i)=\Phi(t,i+1)-\Phi(t,i) \quad {\scriptstyle t=0,1,...,N\ ,\ i=0,1,...} \tag{23}$$

$$\Delta\Phi(t,i)=\frac{1}{2}\left(\varepsilon^2(t+1,i+1)-\varepsilon^2(t+1,i)\right) \tag{24}$$

We can rewrite (18) as:

$$\begin{aligned}\Delta\Phi(t,i)&=\frac{1}{2}\left[\varepsilon(t+1,i+1)+\varepsilon(t+1,i)\right]\left[\varepsilon(t+1,i+1)-\varepsilon(t+1,i)\right]\\&=\Delta\varepsilon(t+1,i)\left[\varepsilon(t+1,i)+\frac{1}{2}\Delta\varepsilon(t+1,i)\right]\end{aligned} \tag{25}$$

Where $\Delta\varepsilon(t+1,i)=\varepsilon(t+1,i+1)-\varepsilon(t+1,i)$ is the error difference that can be substituted by using Taylor series expansion as follows [41-43] :

$$\Delta\varepsilon(t+1,i)=\Delta\underline{w}^T(t,i)\left(\frac{\partial\varepsilon(t+1,i)}{\underline{w}(t,i)}\right)+tr\left[\Delta V_a^T(t,i)\left(\frac{\partial\varepsilon(t+1,i)}{V_a^T(t,i)}\right)\right] \tag{26}$$

where $\Delta\underline{w}(t,i)=\underline{w}(t,i+1)-\underline{w}(t,i)$ and $\Delta V_a(t,i)=V_a(t,i+1)-V_a(t,i)$. Substituting $\Delta\varepsilon(t+1,i)$ from (26) into (25) gives:

$$\Delta\Phi(t,i)=\left(\begin{array}{l}\Delta\underline{w}^T(t,i)\left(\frac{\partial\varepsilon(t+1,i)}{\hat{\underline{w}}(t,i)}\right)+\\ tr\left[\Delta V_a^T(t,i)\left(\frac{\partial\varepsilon(t+1,i)}{V_a^T(t,i)}\right)\right]\end{array}\right)*\left(\begin{array}{l}\varepsilon(t+1,i)+\frac{1}{2}\Delta\underline{w}^T(t,i)\left(\frac{\partial\varepsilon(t+1,i)}{\underline{w}(t,i)}\right)+\\ \frac{1}{2}tr\left[\Delta V_a^T(t,i)\left(\frac{\partial\varepsilon(t+1,i)}{V_a^T(t,i)}\right)\right]\end{array}\right) \tag{27}$$

We consider $\Delta\Phi(t,i)$ as follow:

$$\Delta\Phi(t,i)=\Delta\Phi_1(t,i)+\Delta\Phi_2(t,i)+\Delta\Phi_3(t,i) \tag{28}$$

Where

$$\Delta\Phi_1(t,i)=\left(\Delta\underline{w}^T(t,i)\left(\frac{\partial\varepsilon(t+1,i)}{\underline{\hat{w}}(t,i)}\right)\right)\left(\varepsilon(t+1,i)+\frac{1}{2}\Delta\underline{w}^T(t,i)\left(\frac{\partial\varepsilon(t+1,i)}{\underline{w}(t,i)}\right)\right) \tag{29}$$

$$\Delta\Phi_2(t,i)=\left(tr\left[\Delta V_a^T(t,i)\left(\frac{\partial\varepsilon(t+1,i)}{V_a^T(t,i)}\right)\right]\right)\left(\varepsilon(t+1,i)+\frac{1}{2}tr\left[\Delta V_a^T(t,i)\left(\frac{\partial\varepsilon(t+1,i)}{V_a^T(t,i)}\right)\right]\right) \tag{30}$$

$$\Delta\Phi_3(t,i)=\left(\Delta\underline{w}^T(t,i)\left(\frac{\partial\varepsilon(t+1,i)}{\underline{w}_a(t,i)}\right)\right)\left(tr\left[\Delta V_a^T(t,i)\left(\frac{\partial\varepsilon(t+1,i)}{V_a^T(t,i)}\right)\right]\right) \tag{31}$$

Substituting (11) - (14) into (29) - (31), we have:

$$\Delta\Phi_1(t,i)==\begin{pmatrix}-\alpha_w(i)\left\|\frac{\partial\hat{y}(t+1,i)}{\partial\underline{w}(t,i)}\right\|^2+\\ \frac{1}{2}\alpha_w(i)^2\left\|\frac{\partial\hat{y}(t+1,i)}{\partial\underline{w}(t,i)}\right\|^4\end{pmatrix}\varepsilon^2(t+1,i) \tag{32}$$

$$\Delta\Phi_1(t,i)=\alpha_w(i)\left\|\frac{\partial\hat{y}(t+1,i)}{\partial\underline{w}(t,i)}\right\|^2\left(-1+\frac{1}{2}\alpha_w(i)\left\|\frac{\partial\hat{y}(t+1,i)}{\partial\underline{w}(t,i)}\right\|^2\right)\varepsilon^2(t,i) \tag{33}$$

Also

$$\Delta\Phi_2(t,i)=\alpha_V(i)\left\|\frac{\partial\hat{y}(t+1,i)}{\partial V_a^T(t,i)}\right\|_f^2\left(-1+\frac{1}{2}\alpha_V(i)\left\|\frac{\partial\hat{y}(t+1,i)}{\partial V_a^T(t,i)}\right\|_f^2\right)\varepsilon^2(t+1,i) \tag{34}$$

$$\Delta\Phi_3(t,i)=\alpha_w(i)\alpha_V(i)\left\|\frac{\partial\hat{y}(t+1,i)}{\partial\underline{w}(t,i)}\right\|^2\left\|\frac{\partial\hat{y}(t+1,i)}{\partial V_a^T(t,i)}\right\|_f^2\varepsilon^2(t+1,i) \tag{35}$$

We define $\psi_1(t,i)=\alpha_w(i)\left\|\frac{\partial\hat{y}(t+1,i)}{\partial\underline{w}(t,i)}\right\|^2$ $\psi_2(t,i)=\alpha_V(i)\left\|\frac{\partial\hat{y}(t+1,i)}{\partial V_a^T(t,i)}\right\|_f^2$ and substitute them into (33) – (35), so we have:

$$\Delta\Phi_1(t,i)=\psi_1(t,i)\left(-1+\frac{1}{2}\psi_1(t,i)\right)\varepsilon^2(t+1,i) \tag{36}$$

$$\Delta\Phi_2(t,i)=\psi_2(t,i)\left(-1+\frac{1}{2}\psi_2(t,i)\right)\varepsilon^2(t+1,i) \tag{37}$$

$$\Delta\Phi_3(t,i)=\psi_1(t,i)\psi_2(t,i)\varepsilon^2(t,i) \tag{38}$$

Using (36) - (38) and substituting into (28), we have:

$$\Delta\Phi(t,i)=\frac{1}{2}\left(\psi_1^2(t,i)+\psi_2^2(t,i)\right)+\psi_1\psi_2-\psi_1-\psi_2 \tag{39}$$

$$\Delta\Phi(t,i)=\left(\psi_1(t,i)+\psi_2(t,i)\right)\left(-1+\frac{1}{2}\left(\psi_1(t,i)+\psi_2(t,i)\right)\right) \tag{40}$$

$\Delta\Phi(t,i)$ is negative if

$$-1+\frac{1}{2}\left(\psi_1(t,i)+\psi_2(t,i)\right)<0$$

so the following condition must be satisfied:

$$0<\psi_1(t,i)+\psi_2(t,i)<2 \tag{41}$$

If we design $\alpha_w(i)<\frac{1}{\mu_w^2(i)}$ and $\alpha_V(i)<\frac{1}{\mu_V^2(i)}$ Since $\mu_w(i)$ and $\mu_v(i)$ are defined by (20) and (21), so $\psi_1(t,i)$ and $\psi_2(t,i)$ are: $0<\psi_1(t,i)<1$ and $0<\psi_2(t,i)<1$

Therefore the condition (41) is satisfied, so $\Delta\Phi(t,i)<0$, consequently, the convergence of the proposed neural network is guaranteed.

the proof of theorem 1 is completed.

Remark2. We have , since $h(.)$ is a sigmoidal function therefore $\max_t\left\|\underline{h}\left(V_a^T(t,i)\underline{x}_a(t,i)\right)\right\|=m$ and so $\alpha_w(i)$ is designed $\alpha_w(i)\triangleq\alpha_w<\frac{1}{m^2}$.

4. Iterative learning control based on neural network

A neural network based ILC for system (1) is presented in this section. Since the structure of the proposed neural identifier is affine in control input, so proposed ILC can be easily applied to it. Iterative learning law is proposed as following:

$$\underset{t=0,1,\dots,N\quad i=0,1,\dots}{u(t,i+1)=u(t,i)+\Delta u(t,i)} \tag{42}$$

Which in

$$\Delta u(t,i) = ke(t+1,i) - \underline{w}(t,i+1)^T \left(\Delta \underline{h}_a(t,i)\right) - \Delta \underline{w}^T(t,i)\underline{h}_a\left(V_a^T(t,i)\underline{x}_a(t,i)\right) \quad t=0,1,\ldots,N \quad i=0,1,\ldots \tag{43}$$

$$\Delta \underline{h}_a(t,i) = \underline{h}\left(V_a^T(t,i+1)\underline{x}_a(t,i+1)\right) - \underline{h}\left(V_a^T(t,i)\underline{x}_a(t,i)\right) \tag{44}$$

Where k and $e(t+1,i) = y_d(t+1) - \hat{y}(t+1,i)$ are the learning gain and the tracking error respectively

Theorem 2: Iterative learning control based on the neural network that presented in the previous section and theorem 1, with learning law (42-44) is applied to (2) such that $\hat{y}(t+1,i)$ tracks $y_d(t+1)$. The convergence of the proposed ILC is guaranteed if we design the learning gain k such that $|1-k|<1$.

Proof:

$$e(t+1,i+1) - e(t+1,i) = -\left(\hat{y}(t+1,i+1) - \hat{y}(t+1,i)\right) \tag{45}$$

$$e(t+1,i+1) - e(t+1,i) = -\underline{w}(t,i+1)^T \underline{h}\left(V_a^T(t,i+1)\underline{x}_a(t,i+1)\right) + \underline{w}(t,i)^T \underline{h}\left(V_a^T(t,i)\underline{x}_a(t,i)\right) - \left(u(t,i+1) - u(t,i)\right) \tag{46}$$

After adding and subtracting $\underline{w}(t,i+1)^T \underline{h}\left(V_a^T(t,i)\underline{x}_a(t,i)\right)$ to (46), we have

$$e(t+1,i+1) - e(t+1,i) = -\Delta u(t,i) - \underline{w}(t,i+1)^T \left(\underline{h}\left(V_a^T(t,i+1)\underline{x}_a(t,i+1)\right) - \underline{h}\left(V_a^T(t,i)\underline{x}_a(t,i)\right)\right) - \Delta \underline{w}^T(t,i)\underline{h}_a\left(V_a^T(t,i)\underline{x}_a(t,i)\right) \tag{47}$$

Using learning law (42-44), we have

$$e(t+1,i+1) = (1-k)e(t+1,i) \tag{48}$$

If we design the learning gain k such that $|1-k|<1$ so

$$\left|e(t+1,i+1)\right| < \left|e(t+1,i)\right| \tag{49}$$

According to (49), the convergence of the proposed ILC is guaranteed and the proof of theorem 2 is completed.

By applying the proposed iterative control law to system (1), within a time-limited range, $\hat{y}(t+1,i)$ follows the desired output trajectory $y_d(t+1)$, simultaneously $y(t+1,i)$ follows $\hat{y}(t+1,i)$. Therefore tracking the desired output trajectory $y_d(t+1)$ by $y(t+1,i)$ is guaranteed as the iteration number i tends to infinity, consequently, the control and identification aims are achieved simultaneously.

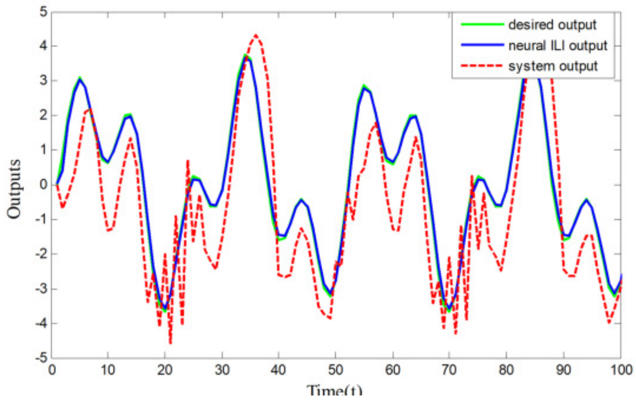

Fig. 3. Tracking and identification performance at the second iteration for the case1

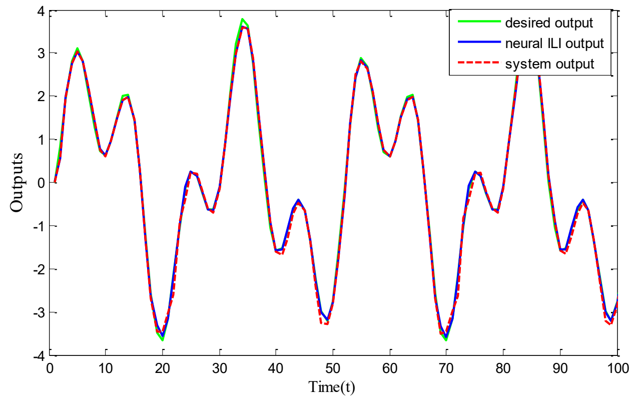

Fig.4. Tracking and identification performance at the iteration $i = 40$ for the case1

5. SIMULATION STUDY

To show the effectiveness of the proposed based neural network ILC, we consider results of the computer simulation an example. The example is done in three different cases to evaluate different aspects of algorithm. We also present a comparison between our proposed method and the neural controller proposed in [39] that mentioned in section 1 (the newest ILC that has been applied to dynamic systems represented by NARMAX models).

Example 1: Consider non-BIBO iterative time-varying NARMAX system [41,44] in three cases. The system is presented as:

$$y(t+1,i) = 0.2y(t-1,i)^2 + 0.2y(t,i) + 0.4\sin\left(0.5\left(y(t,i)+y(t-1,i)\right)\right) \cos\left(0.5\left(y(t,i)+y(t-1,i)\right)\right) + 1.2u(t,i) + \cos(t)$$

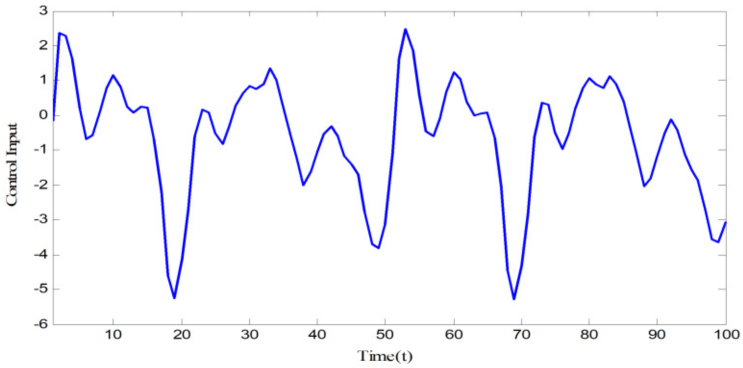

Fig. 5. Control signal $u(t,i)$ at the iteration $i = 40$ for the case1

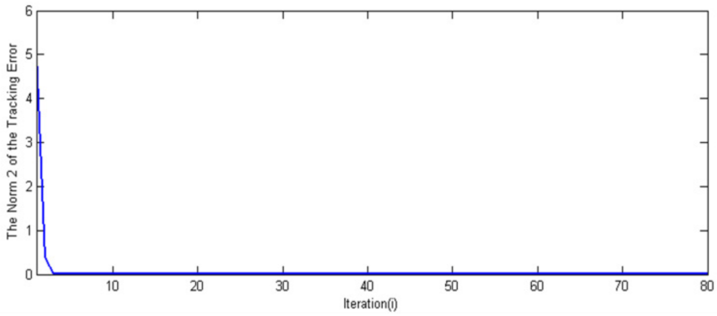

Fig. 6. The norm 2 of the tracking error $e_c(t,i)$ with respect to i for the case1

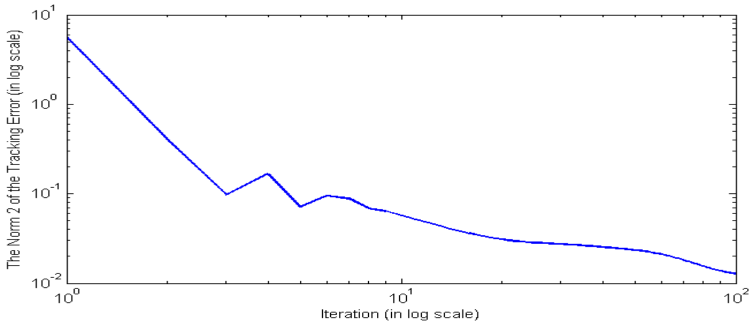

Fig. 7. The norm 2 of the tracking error $e_c(t,i)$ with respect to i, in log scale, for the case1

Which $t \in [0,100]$ and $p = 3$. The desired trajectory is produced by the following system:

$$y_d(t+1) = 0.6 y_d(t) + r(t)$$
$$r(t) = \sin(\frac{2\pi}{25} t) + \sin(\frac{2\pi}{10} t)$$

In case 1, $y(0,i)$ is iteration –invariant as $y(0,i) = 0.5$. We design learning gain $k = 0.9$, neural network gain $\alpha_w = 0.01$, number of hidden neurons $m = 12$. Control input and weights are designed randomly with values in interval $[-1,1]$ at iteration $i = 0$. The trajectories of the desired output $y_d(t+1,i)$ (green-solid), the neural ILI output $\hat{y}(t+1,i)$ (blue-solid) and the system output $y(t+1,i)$ (red-dashed), at 2th and 40th are shown in Figs. 3 and 4 (for case 1). These Figures demonstrate that by increasing the iterations number, the neural network output is rapidly converted to the given desired output trajectory (in the initial iterations), but to converge the system output to the desired output trajectory must firstly identification be reached precisely. Also the control input is shown in Fig. 5. For a more precise evaluation of the convergence rate, the norm 2 of the tracking error $e(t+1,i)$ is shown in Figs. 6. The norm 2 of the tracking error in log scale is shown in Fig. 7. The results of the comparison between the proposed method in this paper and the method presented in [39] is shown in Fig. 8. Elapsed time of simulation of the proposed method is 5.276404 seconds and Elapsed time of simulation of the method presented in [39] is 11.134760. the results of comparison is represented in remarks 3-5.

Remark 3. According to the results of the comparison Fig. 8, the proposed method in this paper has better and

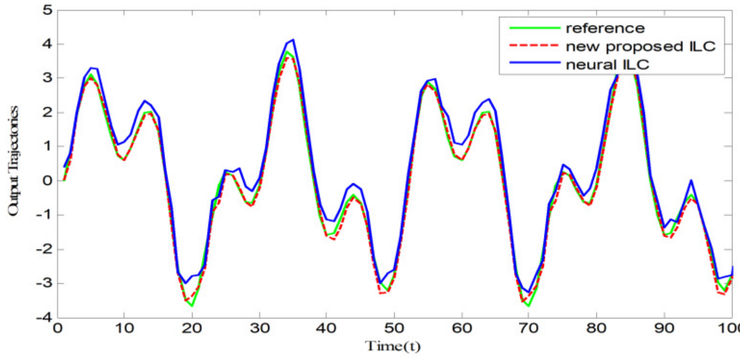

Fig. 8. Reference tracking (green), the proposed ILC (red), the neural controller [39] (blue) , for the case1

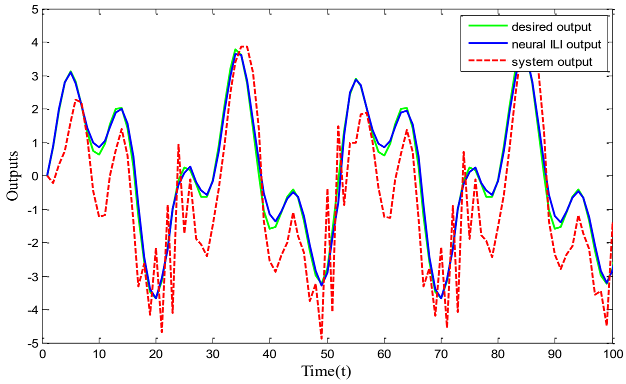

Fig. 9. Tracking and identification performance at the second iteration for the case2

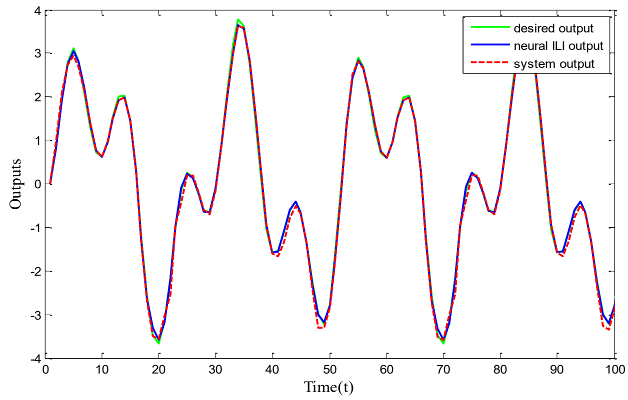

Fig. 10. Tracking and identification performance at the iteration $i = 40$ for the case2

more accurate performance than the method presented in [39]. Also, in the method presented in [39], there is the identification process in every control iteration (identification process and control process are not simultaneously), so it takes more time than our proposed method.

Remark 4. In the method presented in [39], the convergence is not monotonic, so, it must use a law pass Q-filter to achieve the monotonic convergence. While the convergence of tracking error in our proposed ILC is monotonic (see Fig. 6).

In case 2, $y(0,i)$ is iteration –variant as $y(0,i) = \sin(\frac{i\pi}{20})$, we design parameters same as case 1. The trajectories of the desired output $y_d(t+1,i)$ (green-solid), the neural ILI output $\hat{y}(t+1,i)$ (blue-solid) and the system output $y(t+1,i)$ (red-dashed), at 2th and 40th are shown in Figs. 9 and 10. Also the control input is shown in Fig. 13. For a more precise evaluation of the convergence rate, the norm 2 of the tracking error $e(t+1,i)$ is shown in Figs. 11. The norm 2 of the tracking error in log scale is shown in Fig. 12.

The results of case 2 shows that the proposed method in this paper can be applied to the system with iteration –variant initial conditions, whereas the neural controller proposed in [39] is applied to the system returns to the same initial conditions at the start of each trial.

Remark 5. The results of case 2 shows that the proposed method in this paper can be also applied to the system with iteration –variant initial conditions, whereas the neural controller proposed in [39] is only applied to the system with iteration –invariant initial conditions.

In case 3, $y(0,i)$ is iteration –invariant as $y(0,i) = 0.5$. We change learning gain k and design it $k = 0.2$, other parameters are designed same as case1. The trajectories of the desired output $y_d(t+1,i)$ (green-solid), the neural ILI output $\hat{y}(t+1,i)$ (blue-solid) and the system output $y(t+1,i)$ (red-dashed), at 2th and 40th are shown in Figs. 14 and 15 (for case 3). Also the control input is shown in Fig. 16. The norm 2 of the tracking error $e(t+1,i)$ is shown in Figs. 17.

The results of case 3 demonstrate that the neural network output is slower than converted to the given desired output trajectory (at the initial iterations) , but the tracking performance of system output is finally more accurate than case 1.

Remark 6. By decreasing the learning gain k , the speed of the convergence decreases but the accuracy increases.

Remark 7. According to three cases have been represented above, in the proposed ILC based on the proposed neural network, the perfect tracking of the desired output by system output is achieved only after perfect identification.

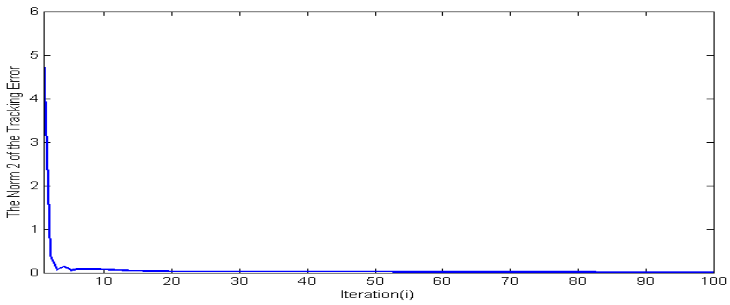

Fig. 11. The norm 2 of the tracking error $e_c(t,i)$ with respect to i, for the case2

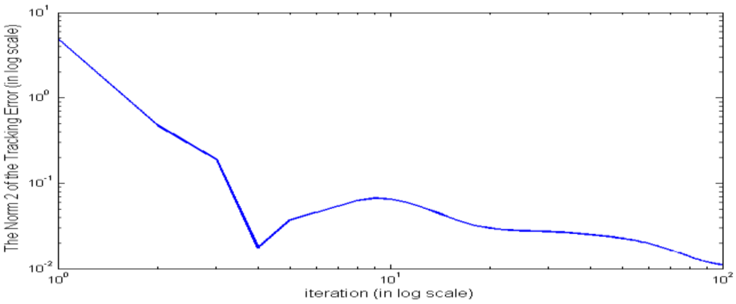

Fig.12. The norm 2 of the tracking error $e_c(t,i)$ with respect to i, in log scale, for the case2

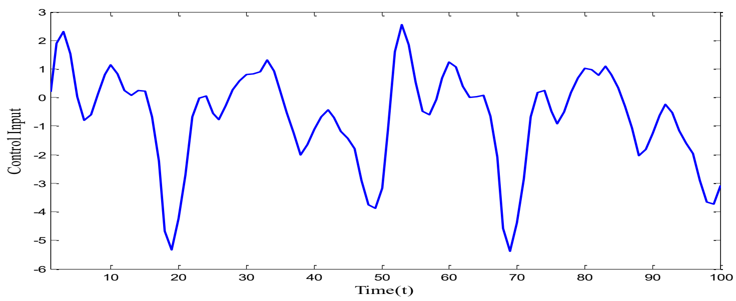

Fig. 13. Control signal $u(t,i)$ at the iteration $i = 40$ for the case2

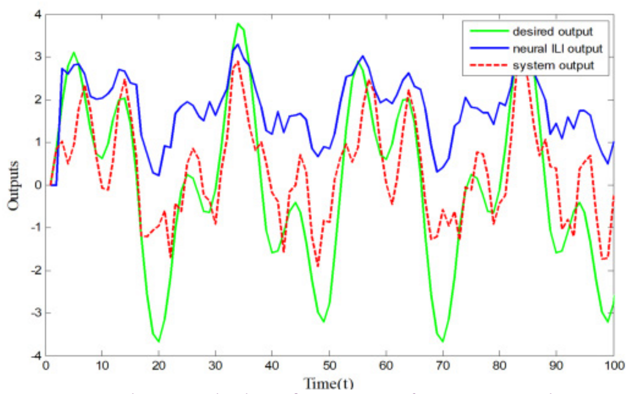

Fig. 14. Tracking and identification performance at the second iteration for the case 3

6. Conclusion

This paper presents a new iterative learning controller is composed of ILI and ILC for a general unknown discrete time-varying nonlinear non-affine system described by NARMAX model. A multi-layer neural network is used for identification. Since the system is time-varying, we propose a time-varying neural network that is affine in the control input. The new proposed ILC based on the proposed neural network applied to the system. Weights of the neural network are updated along the control iteration , so we have ILC and ILI simultaneously, consequently, the proposed approach in comparison with approach that has perfect identification process at each iteration takes less time. The convergence of both the trajectory tracking error and identification error is guaranteed along the iteration domain with repeating the process within a time-limited range. By increasing the iterations number, the neural network output is converted to the given desired output trajectory but to converge the system output to the desired output trajectory must firstly identification be reached precisely. Illustrative example and

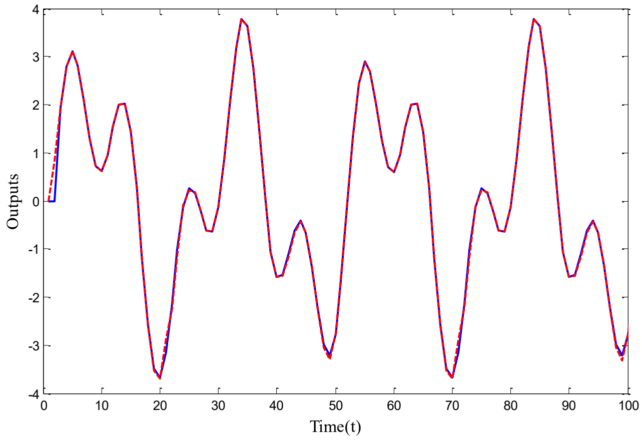

Fig.15. Tracking and identification performance at the iteration $i = 40$ for the case 3

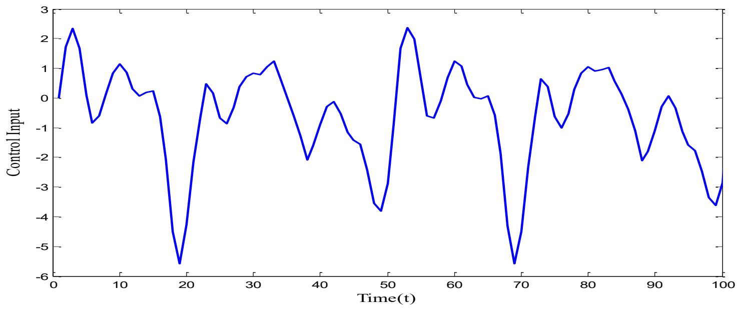

Fig. 16. Control signal $u(t,i)$ at the iteration $i = 40$ for the case 3

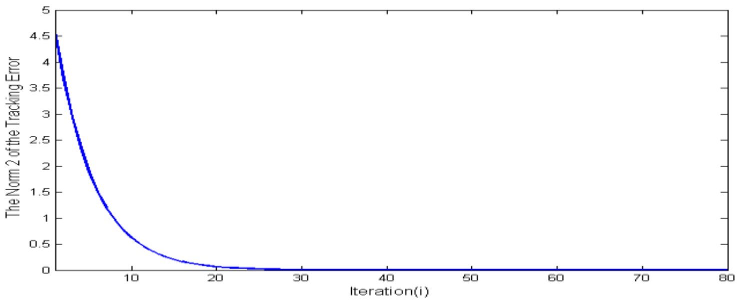

Fig. 17. The norm 2 of the tracking error $e_c(t,i)$ with respect to i for the case3

results of the comparison show to be useful the proposed method in this paper. It should be mentioned that the system initial condition can be iteration variant in the proposed controller , whereas in most ILC approaches the system initial condition are iteration-invariant.

This paper consider discrete time nonlinear non-affine system, future studies could consider nonlinear non-affine continuous system.

References

[1] S. Arimoto, S. Kawamura, F. Miyazaki, Bettering operation of robots by learning, Journal of Robotic systems, 1(2) (1984) 123-140.
[2] A. Tayebi, S. Abdul, M. Zaremba, Y. Ye, Robust iterative learning control design: application to a robot manipulator, IEEE/ASME Transactions on mechatronics, 13(5) (2008) 608-613.
[3] L. Zhang, S. Liu, Iterative learning control for flexible manipulator using fourier basis function, International Journal of Automation and Computing, 12(6) (2015) 639-647.
[4] W. Hoffmann, K. Peterson, A.G. Stefanopoulou, Iterative learning control for soft landing of electromechanical valve actuator in camless engines, IEEE Transactions on control systems technology, 11(2) (2003) 174-184.
[5] T. Liu, F. Gao, Y. Wang, IMC-based iterative learning control for batch processes with uncertain time delay, Journal of Process Control, 20(2) (2010) 173-180.
[6] J. Chani-Cahuana, P.N. Landin, C. Fager, T. Eriksson, Iterative learning control for RF power amplifier linearization, IEEE Transactions on Microwave Theory and Techniques, 64(9) (2016) 2778-2789.
[7] H.-S. Ahn, Y. Chen, K.L. Moore, Iterative learning control: Brief

survey and categorization, IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 37(6) (2007) 1099-1121.
[8] J. Han, D. Shen, C.-J. Chien, Terminal Iterative Learning Control for Discrete-Time Nonlinear System Based on Neural Networks, Journal of the Franklin Institute, 355 (2018) 3641–3658.
[9] M. Kim, T.-Y. Kuc, H. Kim, J.S. Lee, Adaptive iterative learning controller with input learning technique for a class of uncertain MIMO nonlinear systems, International Journal of Control, Automation and Systems, 15(1) (2017) 315-328.
[10] A. Madady, An extended PID type iterative learning control, International Journal of Control, Automation and Systems, 11(3) (2013) 470-481.
[11] A. Madady, H.R. Reza-Alikhani, A guaranteed monotonically convergent iterative learning control, Asian Journal of Control, 14(5) (2012) 1299-1316.
[12] A. Madady, H.-R. Reza-Alikhani, S. Zamiri, Optimal N-Parametric Type Iterative Learning Control, International Journal of Control, Automation and Systems, 16(5) (2018) 2187-2202.
[13] K. Wan, X.-D. Li, Iterative learning control for two-dimensional linear discrete systems with Fornasini-Marchesini model, International Journal of Control, Automation and Systems, 15(4) (2017) 1710-1719.
[14] J.-X. Xu, Y. Tan, A composite energy function-based learning control approach for nonlinear systems with time-varying parametric uncertainties, IEEE Transactions on Automatic Control, 47(11) (2002) 1940-1945.
[15] S. Yang, J.X. Xu, D. Huang, Y. Tan, Synchronization of heterogeneous multi-agent systems by adaptive iterative learning control, Asian Journal of Control, 17(6) (2015) 2091-2104.
[16] N. Liu, A. Alleyne, Iterative learning identification for linear time-varying systems, IEEE Transactions on Control Systems Technology, 24(1) (2016) 310-317.
[17] N. Liu, Learning identification and control for repetitive linear time-varying systems, University of Illinois at Urbana-Champaign, 2014.
[18] N. Liu, A. Alleyne, Iterative Learning Identification/Iterative Learning Control for Linear Time-Varying Systems, Journal of Dynamic Systems, Measurement, and Control, 138(10) (2016) 101005.
[19] S. Billings, Identification of nonlinear systems using parameter estimation techniques, in: Institute of Electrical Engineers Conference, 1981, pp. 183-190.
[20] S. Billings, W. Voon, Least squares parameter estimation algorithms for non-linear systems, (1984).
[21] S. Chen, S.A. Billings, Representations of non-linear systems: the NARMAX model, International journal of control, 49(3) (1989) 1013-1032.
[22] E. Camporeale, S. Wing, J. Johnson, Machine learning techniques for space weather, Elsevier, 2018.
[23] C.C. Huang, C.H. Loh, Nonlinear identification of dynamic systems using neural networks, Computer-Aided Civil and Infrastructure Engineering, 16(1) (2001) 28-41.
[24] J. Yan, J.R. Deller Jr, NARMAX model identification using a set-theoretic evolutionary approach, Signal Processing, 123 (2016) 30-41.
[25] X.C. Guan, D.Y. Zhao, Q.M. Zhu, NARMAX modelling and U-model control design for continuous stirred tank reactor (CSTR), in: 2016 35th Chinese Control Conference (CCC), IEEE, 2016, pp. 1964-1969.
[26] M.B. Menhaj, computational intelligence: fundamentals of neural networks, Amir Kabir University Publishing, tehran, iran, (2017) .
[27] M.B. Menhaj, adaptive control systems, Amir Kabir University Publishing, tehran,iran, (2017).
[28] H. Aliyari, S. Hosseinian, H. Sahraei, M. Menhaj, Effect of proximity to high-voltage fields: results of the neural network model and experimental model with macaques, International Journal of Environmental Science and Technology, (2018) 1-12.
[29] B. Karimi, M.B. Menhaj, Non-affine nonlinear adaptive control of decentralized large-scale systems using neural networks, Information Sciences, 180(17) (2010) 3335-3347.
[30] F. Abedini, M.B. Menhaj, M.R. Keyvanpour, An MLP-based representation of neural tensor networks for the RDF data models, Neural Computing and Applications, (2017) 1-10.
[31] C.J. Chien, L.C. Fu, An iterative learning control of nonlinear systems using neural network design, Asian Journal of Control, 4(1) (2002) 21-29.
[32] C. Fu, M. Poch, Application of a multi-layered neural network to system identification, International journal of systems science, 24(8) (1993) 1601-1609.
[33] S. Chen, S. Billings, P. Grant, Non-linear system identification using neural networks, International journal of control, 51(6) (1990) 1191-1214.
[34] D. Xu, Z. Li, W. Wu, X. Ding, D. Qu, Convergence of gradient descent algorithm for diagonal recurrent neural networks, in: 2007 Second International Conference on Bio-Inspired Computing: Theories and Applications, IEEE, 2007, pp. 29-31.
[35] D. Xu, H. Zhang, D.P. Mandic, Convergence analysis of an augmented algorithm for fully complex-valued neural networks, Neural Networks, 69 (2015) 44-50.
[36] C. Shao, J. Nie, F. Gao, A Robust Iterative Learning Control with Neural Networks for Robot, IFAC Proceedings Volumes, 37(1) (2004) 779-784.
[37] X. Yu, Z. Hou, C. Yin, Iterative learning control for discrete-time nonlinear systems based on adaptive tuning of 2D learning gain, in: 2017 36th Chinese Control Conference (CCC), IEEE, 2017, pp. 3581-3586.
[38] K. Patan, M. Patan, D. Kowalów, Neural networks in design of iterative learning control for nonlinear systems, IFAC-PapersOnLine, 50(1) (2017) 13402-13407.
[39] K. Patan, M. Patanl, Design and convergence of iterative learning control based on neural networks, in: 2018 European Control Conference (ECC), IEEE, 2018, pp. 1-6.
[40] W.-L. Yan, M.-X. Sun, Identification of discrete-time varying nonlinear systems using time-varying neural networks, in: 2010 8th World Congress on Intelligent Control and Automation, IEEE, 2010, pp. 301-306.
[41] C.-C. Ku, K.Y. Lee, Diagonal recurrent neural networks for dynamic systems control, IEEE transactions on neural networks, 6(1) (1995) 144-156.
[42] C.-H. Lee, C.-C. Teng, Identification and control of dynamic systems using recurrent fuzzy neural networks, IEEE Transactions on fuzzy systems, 8(4) (2000) 349-366.
[43] W.T. Miller, P.J. Werbos, R.S. Sutton, Neural networks for control, MIT press, 1995.
[44] Y.C. Wang and C. J. Chien, An output-recurrent-neural-network-based iterative learning control for unknown nonlinear dynamic plants, Journal of control science and engineering, 2012 (2012), 1-9.

HOW TO CITE THIS ARTICLE

F. Afsharnia, A. Madady, M.B. Menhaj, Iterative learning identification and control for dynamic systems described by NARMAX model, AUT J. Model. Simul., 51(2) (2019) 199-210.

DOI: 10.22060/miscj.2019.16029.5151