

System Identification of Thermal Process using Elman Neural Networks with No Prior Knowledge of System Dynamics

Ibraheem Kasim Ibraheem
Electrical Engineering Dept.
College of Engineering, Baghdad University
Baghdad, Iraq

ABSTRACT

Dynamic Neural networks have been verified as identifiers due to its capability for manipulating processes in parallel and enduring noisy sorts of the input signals. They make them outstanding contenders for system identification applications. This paper presents a method for a black box system identification based on Elman neural network (ENN) for thermal process system to generate a prototype for the dynamical system of the thermal process without any past information about the system dynamics. This identification approach is compared to its counterpart conventional feedforward neural network (CFFNN) based system identification. The comparative simulations show that the ERNN gives an excellent results and outperforms the CFFNN in terms of accuracy with little degradation in the speed of convergence which make this neural network a motivating candidate for adaptive and gain scheduling controllers.

General Terms

system identification, Recurrent neural networks, thermal process, feedforward neural network, mean square error.

Keywords

Backpropagation algorithm, Elman neural networks, black box modeling, online training, offline training

1. INTRODUCTION

The primary objective of system identification is to set up a sufficient dynamical model for the system with the end goal that the parameters, which give the best fit between the computed the model-produced values and input–output information, are estimated, the model parameters and structure might be identified [1]. It is the first step in the indirect neural control design. In this case, the controller itself may not be a neural network, but it is derived from a plant that is modeled by a neural network. The idea is that for the identification process to produce a model of the dynamical system without any prior knowledge about the system's dynamics. This is referred to as black box or nonparametric system identification [2]. It is advantageous when the engineer is provided with the system's impulse or step response.

Thermal processes are those systems that include the exchange of heat from the individual body to another. When the temperatures of the body and its surroundings are not the same, heat transfer will begin, starting from the body with higher temperature toward the body of the lower temperature one, complying with the energy conservation law. Instances of temperature systems are refrigerators, radiators, ventilation systems, to mention just a few. [3].

Artificial Neural Networks (ANNs) represent one of the most successful identification techniques used to model nonlinear dynamical systems. This is due to their ability to learn by examples associated with inherent robustness and nonlinear characteristics [4]. The ANN is a profoundly interconnected system of a huge number of managing units named neurons in a structure stimulated by the brain. An NN can be massively parallel and therefore is said to exhibit parallel distributing processing. There has been a huge writing on NNs in the most recent decades, whose start was maybe set apart by the main IEEE Worldwide Conference on Neural Systems in 1987. It has been perceived that NN presents various possible advantages for applications in the field of designing control systems, especially to model nonlinear processes. Some attractive characteristics of NN are its capability for “*learning via examples.*” They don't need an earlier information. This features enabled NN to approximate well any nonlinear continuous functions [5]. But feedforward neural networks models suffer from slow convergence, local minimum, overfitting, have high computational cost and need a large number of iterations for its training due to the availability of hidden layer [6]. In black box system identification, however, the really important task is to build models for dynamic systems. In dynamic systems, the output at a given time instant depends not only on its current inputs but the previous behavior of the system. Dynamic systems are systems with memory. There are several ways to form dynamical NNs based on static neurons, nevertheless in all ways storage elements and/or apply feedback will be used. Both methods can produce several different dynamic NN architectures [7]. It is well known that the recurrent neural networks (RNNs) usually provide a smaller architecture (the exogenous noise signals have little influence on these kinds of networks) than most of the non-recursive NNs like multilayer perceptron, Radial basis function neural networks (RBFNN). Also, their feedback properties (they have dynamic memories) make them dynamic and more efficient to represent nonlinear systems precisely which are essential for nonlinear forecasting and time-series estimation. RNNs have correctly modeled many of the Autoregressive Moving Average (ARMA) processes for nonlinear dynamical system identification [6].

The first step in solving system identification problem is to select an appropriate class of model structure. The models can be characterized in many different ways, such as parametric (e.g. state space models) and nonparametric models (e.g. impulse response) [8].

There has been a lot of research done in the field of system identification using artificial neural networks. In [1, 6, 9] swarm optimizations algorithms like PSO, Differential evolution and Electromagnetism-like respectively have been

considered in the system identification of the thermal process and time series prediction. While a new class of nonlinear fractional models based on the Volterra series is proposed for modeling such nonlinear diffusive phenomena in [10]. A sparse Bayesian and its associated algorithm have been proposed in [11] to identify the system nonlinear functional forms and their associated parameters from a limited number of time-series data points. For this, Pan W. *et al.* cast this identification problem as a sparse linear regression problem and took a Bayesian viewpoint to solve it. The works in [8, 12] have made use of spline techniques for the design of adaptive filters and used for linear and nonlinear system identification. Parametric and nonparametric methods using neural feedforward networks have been adopted for the system identification of an experimental turbojet engine in [13]. A parameter identification method is proposed in [14], which makes use of limited measurement information to estimate the parameters of a photovoltaic array model and a sequential quadratic programming method for the problem is introduced. NARX model together with Kalman filter extended Kalman filter, and unscented Kalman filter has been exploited in [15] for identification and estimation of temperature in nitration process. The researchers in [16] developed a fully automated recurrent neural network (FARNN) that is capable of self-structuring its network in a minimal representation of satisfactory performance for unknown dynamic system identification and control. Recurrent neural networks (RNN) have been utilized in system identification for both linear and nonlinear dynamical systems [17, 18]. A particular type of RNN is the Elman recurrent neural networks (ERNN) has been considered in system identification of highly nonlinear dynamical systems, like Twin rotor system [5] and motor rotor resistance [19]. Identification and control of time-delay systems using wavelet neural networks have been achieved in [20]. Finally, [21] presents a comparison study of four recurrent neural networks tuned with a genetic algorithm.

This paper tries to present a method for system identification using Elman neural networks and the training algorithm for this type of neural networks. Also, the obtained results of the identification for thermal process systems using ERNN has been discussed. The data employed in the identification process are collected from the thermal process by firstly linearizing the mathematical model and obtaining the linearized state-space model. Then, the linearized state-space model is converted into constant coefficient difference equation. The validity of the proposed method for system identification using ERNN model is then verified by comparing it to conventional Feedforward Neural Networks (CFNN) based system identification.

The rest of the paper is classified as follows: Section II introduces a brief survey on recurrent neural networks with emphasis on Elman type, the training algorithm is also discussed. The mathematical model of the thermal process is explained in section III. The simulations results with discussions are presented in Section IV. Finally, the conclusions are given in section V.

2. SYSTEM IDENTIFICATION DESIGN

For processes for which experimental input–output data are available, system identification techniques have been shown to be quite effective in deriving a good dynamic representation for the system. The objective in nonparametric identification is to acquire an *i/p-o/p* interpretation for which the dynamics are earlier known to be time invariant and. The strategy is known as nonparametric since it does not include a

parameter vector to be used in searching for the best dynamic model [22]. The nonparametric, time-domain method is useful when the designer has access to the step or impulse response of the system.

The following figure (Fig. 1) illustrates the general issue of the dynamic system identification. The basic principle of the identification is to identify the coefficients of the model as it varies over time so that the error between the model *o/p* and the plant *o/p* has to be minimized [23].

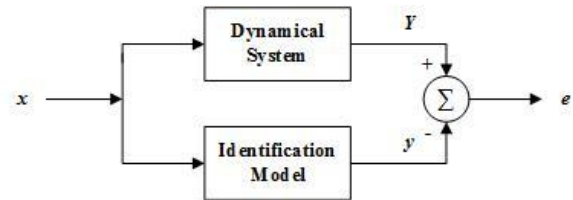


Fig 1: System Identification model.

The principle is for the system identification to deliver a model of the dynamic system without any earlier information about the dynamics of the system, this is called black-box modeling [23]. In black box modeling the model structure does not reflect the structure of the physical system, thus the elements of the model structure have no physical meaning. Instead, such model structure has to be chosen that is flexible enough to represent a large class of systems [7].

The most common measure of discrepancy is the squared error between the model outputs and the observations, which is often called empirical risk [7]:

$$MSE = \frac{1}{2} \sum_{i=1}^N (Y(i) - y(i))^2 \quad (1)$$

System identification based on black box modeling is similar to the general identification case, except that there is no other knowledge about the system than the observations [7]:

$$Z^P = \{x^i, Y^i\}_{i=1}^P$$

A black box model will give a relationship between the observed inputs and outputs. There are many possible techniques which can be applied successfully in system identification (nonlinear function approximation). For example, one can form polynomial functions, when the mapping of the system is approximated by a polynomial, or we complex exponentials can be used, which means, that the mapping of the system is approximated by a Fourier series. But Taylor expansion, wavelets or Volterra series can also be applied. Among the black box structures, dynamic neural networks play an important role [7].

3. SYSTEM IDENTIFICATION USING ELMAN NEURAL NETWORKS

There two kinds of structures of the neural-identification, i.e., serial-parallel and parallel Static neural systems are regularly utilized in serial-parallel structure, and their convergence and stability are effectively ensured [24]. Nonetheless, for the serial-parallel identification arrangement, the system engineer should know precisely what number of past *i/p* and *o/p* data of the system that must be provided in the neural system through a tapped delay line. This is unrealistic under specific conditions when no earlier data of the system to be identified can be acquired [25]. In this section, a dynamic system identification of parallel structure will be setup which depends on the recurrent neural networks RNN. The identification configuration is illustrated in Fig 2.

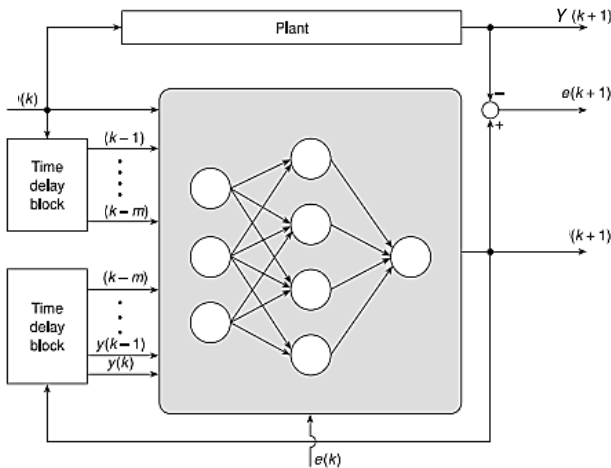


Fig 2: Parallel scheme neural network identification [22].

The architectures of the RNN spans the range from partially connected neural networks to fully interconnected ones, including multilayer perceptron neural networks with separate input and output layers. There are no distinctive layers of nodes for the inputs of Fully connected RNN, and every node has a contribution from every single node in the networks. There might be feedback from an individual node to itself. The nodes in the partially RNNs can be classified as a feedforward nodes, where they are part of the feedforward network structure, and context nodes which serve as a sequential context and receive feedback from other nodes. Weights that represents the connection between the context units and hidden layer node update in the same procedure of the o/p or i/p layer nodes, e.g., backpropagation training algorithm. The context layer nodes units accept a delayed version by one instant of time from the hidden layer nodes. Elman Recurrent Neural Network (ERNN) is One of the well-known partially RNNs [26]. In this kind of networks, the feedback starts from the hidden layer towards the context layer. This approach gives careful consideration to the succession of input information. Fig. 3 delineates the major Elman network which consists of three layers of nodes.

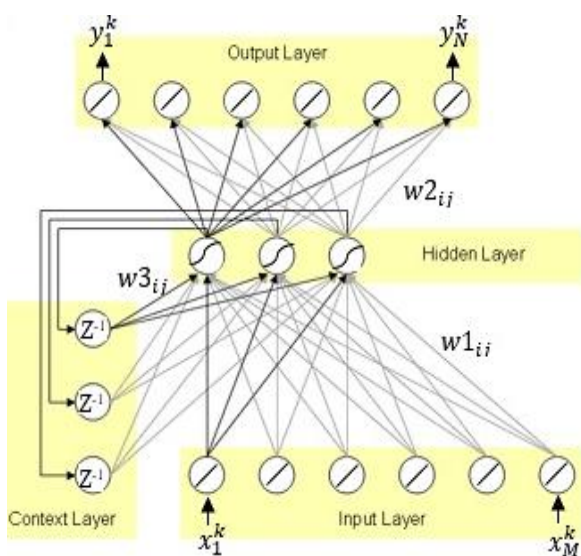


Fig 3: Elman Neural Network structure.

Two different groups of neurons at the starting layer of ERNN differentiate it from other kinds of neural networks. These

groups are the context nodes (internal input neurons) group and the exogenous i/p nodes group. The context nodes are otherwise called as memory nodes due to delay element in this section of the i/p layer which enables the context nodes to store the previous o/p of the hidden layer nodes. When the backpropagation (BP) training algorithm is used to train the ERNN weights, the values of the connection weights between the context units and the hidden layers units are set constant, and typically their values are fixed at 1.0 [22, 23, 25]. Supposedly an ERNN is capable of demonstrating any dynamic system of n^{th} order [26]. For an arbitrary unit in a general recurrent network, the activation at a given time k is given as [22]:

$$o_i(k) = f(\text{tot}_i(k-1)) = f(\sum_j w_{ij} o_j(k-1) + x_i(k-1)) \quad (2)$$

In particular, $o_i(k)$ is the current output signal of node i , $\text{tot}_i(k-1)$ is the net signal input to node i at time “ $k-1$ ”, $o_j(k-1)$ is the output signal of the node j at previous time sample, and w_{ij} is the connection weight between node i and node j . The computation of o_i requires the knowledge of the activations of all units in the *posterior* set, which consists of nodes whose output was relayed to unit i . When computing errors, the errors of all units in the *anterior* set of nodes will have to be determined beforehand. The capability of storing and processing *context information* or *input history* allows them to solve problems that cannot be solved by feedforward networks. However, this comes at the expense of more complex learning and often longer training time [22]. This is discussed in the next section.

The estimated future value of the output $y_p(k+1)$ in the parallel model is expressed as:

$$y_p(k+1) = NNI(\psi, \phi(k)) \quad (3)$$

where NNI stands for mapping provided by the neural network identifier, ψ is the parameter vector which consists of all weights of the dynamic neural network, and $\phi(k)$ signifies the regression vector. The regression vector can be formed as follows [22]:

$$\phi(k) = [y_p(k), \dots, y_p(k-n+1); x(k), \dots, x(k-m+1)] \quad (4)$$

This model is illustrated in Figure 2. and uses the delayed recursions of the established output as some of its input. The corresponding structure is the *Network Output Error (NOE)* model. In an NOE model, there is feedback from model output to its input, so this is a recurrent network. Sometimes NOE model is called as a parallel model [21]. Because of its recurrent architecture, severe instability problem may arise, which cannot be easily handled.

4. TRAINING ELMAN RECURRENT NEURAL NETWORKS

For dynamic training networks, some additional challenges must be solved. Dynamic networks are sequential networks, which means that they implement a nonlinear mapping between input and output data sequences [7]. So the training samples of input–output data pairs of static networks are replaced by input–output data sequences and the goal of the training is to reduce a squared error derived from the elements of the corresponding error sequences. If $\varepsilon(k) = Y(i) - y(i)$ is the output error of a dynamic network at discrete time step k , the squared total error can be defined as [7, 25]:

$$E_{cum} = \frac{1}{2} \sum_{k=1}^K \sum_{i=1}^N (Y_i(k) - y_i(k))^2 \quad (5)$$

Where K is the length of the training sequence, N is the size of the o/p nodes, $Y(i)$ is the plant output at time k , $y(i)$ is the ENN output of the at time k . The following table describes the symbols used in this section.

Table 1. Notations used in deriving the training formulas of the Elman Recurrent Neural Networks (ERNN)

Symbol	Description
$w_{i,j}^1$	The weight that links node i in the i/p layer to node j in the hidden layer
$w_{i,j}^2$	The weight that links node i in the hidden layer to node j in the o/p layer
$w_{i,j}^3$	The weight that links node i in the context nodes group layer to node j in the hidden layer
$toth_i$	the sum of all signals reaching at hidden layer node i
$outh_i$	the o/p of the hidden layer node i
$totc_i$	the sum of all signals reaching context unit i
$outc_i$	the o/p of the context node i
$f(x)$	Activation function of the nodes in the hidden layer with log-sigmoid c/c, i.e. $f(x) = \frac{1}{1 + \exp(-x)}$
$f'(x)$	The derivative of the activation function $f(x)$ and given as $f'(x) = f(x)(1 - f(x))$
M	The size of the i/p to Elman neural networks
N	Size of the o/p layer of the Elman neural network
L	Number of the hidden layer node of ERNN
Z^{-1}	Size one delay element

It is necessary to mention that a linear activation functions have been used in the o/p layer nodes of the ERNN, which sums the signals fed to it. The i/p and o/p nodes interact with the outside environment, whereas the hidden layer and context nodes do not. The context nodes are used only to memorize the previous activations of the hidden nodes and can be considered as one step unit delay element (Z^{-1}) [22, 25]. The weights of the connection between the hidden layer nodes and context nodes are fixed, that is why ENN are sometimes called partially recurrent neural networks [27].

Looking back at Fig. 3, one can write the following equations [25, 27],

$$outc_i(k) = totc_i(k) \quad (6)$$

$$totc_i(k) = outh_i(k-1) \quad (7)$$

$$toth_j(k) = \sum_{i=1}^M w_{i,j}^1(k-1) x_i(k) + \sum_{i=1}^L w_{i,j}^3(k-1) outc_i(k) \quad (8)$$

$$outh_i(k) = f(toth_i(k)) \quad (9)$$

$$y_j(k) = \sum_{i=1}^L w_{i,j}^2(k-1) outh_i(k) \quad (10)$$

Note that the o/p of the context node i at the k^{th} iteration is the one-step delayed o/p of the corresponding hidden node i . It is the existence of the context nodes and the weights connecting them with the hidden nodes that makes the ERNN to be distinguished from other kinds of feedforward neural networks [25].

Let $Y_i(k)$ be the o/p of the plant which is the desired o/p of the ERNN at the k^{th} iteration. To update $w_{i,j}^1$, $w_{i,j}^2$, and $w_{i,j}^3$ at every iteration step, the error E_{cum} given by (5) must be minimized, that is

$$w_{i,j}^S(k) = w_{i,j}^S(k-1) + \mu \Delta w_{i,j}^S(k), \quad S = 1, 2, 3 \quad (11)$$

Where μ is the learning rate, $\Delta w_{i,j}^S(k)$ is given as

$$\Delta w_{i,j}^S(k) = -\frac{\partial E_{cum}(k)}{\partial w_{i,j}^S(k-1)}, \quad S = 1, 2, 3 \quad (12)$$

To update $w_{i,j}^2$ at the k^{th} iteration,

$$\begin{aligned} \frac{\partial E_{cum}(k)}{\partial w_{i,j}^2(k-1)} &= \frac{\partial E_{cum}(k)}{\partial y_i(k)} \cdot \frac{\partial y_i(k)}{\partial tot h_i(k)} \cdot \frac{\partial tot h_i(k)}{\partial w_{i,j}^2(k-1)} \\ &= -(Y_i(k) - y_i(k)) * 1 * outh_i(k) \end{aligned} \quad (13)$$

The update formula for $w_{i,j}^1$ can be derived as follows,

$$\begin{aligned} \frac{\partial E_{cum}(k)}{\partial w_{i,j}^1(k-1)} &= \frac{\partial E_{cum}(k)}{\partial y_i(k)} \cdot \frac{\partial y_i(k)}{\partial out h_i(k)} \cdot \frac{out h_i(k)}{\partial tot h_i(k)} \cdot \frac{\partial tot h_i(k)}{\partial w_{i,j}^1(k-1)} \\ &= \left(-\sum_{i=1}^N (Y_i(k) - y_i(k)) \cdot w_{i,j}^2(k-1) \right) \cdot f'(toth_i(k)) \cdot x_i(k) \end{aligned} \quad (14)$$

Using the derivative chain of rules to derive the update formula for $w_{i,j}^3$, one gets

$$\frac{\partial E_{cum}(k)}{\partial w_{i,j}^3(k-1)} = \frac{\partial E_{cum}(k)}{\partial out h_j(k)} \cdot \frac{\partial out h_j(k)}{\partial w_{i,j}^3(k-1)} \quad (16)$$

At the k^{th} iteration,

$$\begin{aligned} \frac{\partial E_{cum}(k)}{\partial outh_j(k)} &= \frac{\partial E_{cum}(k)}{\partial y_i(k)} \cdot \frac{\partial y_i(k)}{\partial outh_j(k)} \\ &= -\sum_{i=1}^N (Y_i(k) - y_i(k)) \cdot w_{i,j}^2(k-1) \end{aligned} \quad (17)$$

$$\begin{aligned} \frac{\partial outh_i(k)}{\partial w_{i,j}^3(k-1)} &= \frac{\partial outh_j(k)}{\partial toth_j(k)} \cdot \frac{\partial toth_j(k)}{\partial w_{i,j}^3(k-1)} \\ &= f'(toth_j(k)) \cdot outh_i(k-1) \end{aligned} \quad (18)$$

The update formulas for the three sets of weights $w_{i,j}^1$, $w_{i,j}^2$, and $w_{i,j}^3$ are given in tables II and III.

Table 2. Online training for ERNN

Online training
$w_{i,j}^S(k) = w_{i,j}^S(k-1) + \mu \Delta w_{i,j}^S(k), \quad S = 1, 2, 3$
$\Delta w_{i,j}^1(k) = outh_i(k)(1 - outh_i(k))x_i(k) \sum_{i=1}^N (Y_i(k) - y_i(k))w_{i,j}^2(k-1)$
$s\Delta w_{i,j}^2(k) = (Y_i(k) - y_i(k)) outh_i(k)$
$\Delta w_{i,j}^3(k) = f'(toth_j(k)) \cdot outh_i(k-1) \sum_{i=1}^N (Y_i(k) - y_i(k))w_{i,j}^2(k-1)$

Table 3. Offline training for ERNN

Offline training
$w_{ij}^s(k) = w_{ij}^s(k-1) + \mu \Delta w_{ij}^s(k), \quad s = 1, 2, 3$
$\Delta w_{ij}^1 = \sum_{k=1}^K \text{outh}_i(k)(1 - \text{outh}_i(k))x_i(k) \sum_{i=1}^N (Y_i(k) - y_i(k))w_{ij}^2(k-1)$
$\Delta w_{ij}^2 = \sum_{k=1}^K (Y_i(k) - y_i(k)) \text{outh}_i(k)$
$\Delta w_{ij}^3 = \sum_{k=1}^K \hat{f}(\text{toth}_j(k)) \cdot \text{outh}_i(k-1) \sum_{i=1}^N (Y_i(k) - y_i(k))w_{ij}^2(k-1)$

It is worthy to mentions that $\text{outh}_i(0)$ ($i = 1, 2, \dots, L$) are the initial states of the ERNN. They should be specified by the designer of the neural network or may be shortly presume $\text{outh}_i(0) = 0$ ($i = 1, 2, \dots, L$).

5. THERMAL PROCESS MODELING

Thermal processes are those that incorporate the exchange of heat from one body to its surrounding or another body. The energy conservation, heat resistance and capacitance, and heat capacitance which together shape the premise of thermal process modeling will be presented. There are three different means heat can transfer from one material to another one: conduction, convection, and radiation. The first two terms are considered in this paper since radiation heat exchange is significant only if the heat of the emitting body is very high in comparison with the receiving body [3, 28].

For conduction or convection heat transfer,

$$q = Q \Delta T \quad (19)$$

Where q is the heat flow rate, kcal/sec, ΔT temperature difference, C° , Q is a coefficient, kcal/(sec C°). the thermal resistance R for heat transfer between two substances may be defined as follows,

$$R = \frac{\text{change in temperature difference, } C^\circ}{\text{change in heat flow rate, kcal/sec}} = \frac{d(\Delta T)}{dq} = \frac{1}{Q} \quad (20)$$

For either convection or conduction, the thermal resistance is of constant value since the thermal coefficient Q is constant. On the other hand, the thermal capacitance is defined as,

$$C = \frac{\text{change in heat stored, kcal}}{\text{change in temperature}} = mc \quad (21)$$

Where m is the mass of the substance considered, kg and c are the specific heat of a substance, kcal/(kg C°).

Now consider a room with the heater as shown in Fig. 4,

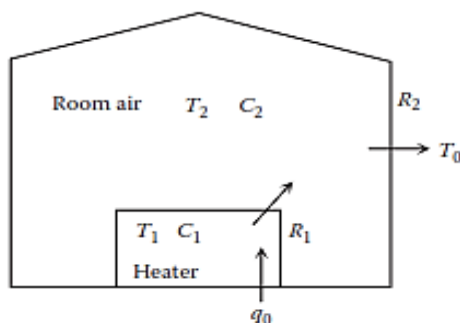


Fig 4: A room with heater [3].

The room has a heater with heat flow rate input of q_o . The thermal capacities of the heater and the room are C_1 and C_2 , respectively. The thermal resistances of the heater-air interface and the room wall-ambient air interface are R_1 and R_2 , respectively. The temperatures of the heater and the room air are T_1 and T_2 . The temperature outside the room is T_o , which is assumed to be constant. Employing energy conservation law to the heating system [3, 28]

$$\frac{dU}{dt} = q_{hi} - q_{ho} \quad (22)$$

Where U is the internal thermal energy and can be expressed as,

$$\frac{dU}{dt} = C_1 \frac{dT_1}{dt}$$

$$q_{hi} = q_o$$

$$q_{ho} = \frac{T_1 - T_2}{R_1}$$

Substituting these expressions in (22) gives

$$C_1 \frac{dT_1}{dt} + \frac{1}{R_1} T_1 - \frac{1}{R_1} T_2 = q_o \quad (23)$$

Same result can be obtained to the room air [3]

$$C_2 \frac{dT_2}{dt} - \frac{1}{R_1} T_1 + \left(\frac{1}{R_1} + \frac{1}{R_2}\right) T_2 = \frac{1}{R_2} T_o \quad (24)$$

The above set of differential equations can be rewritten in state-space of second-order as follows:

$$\begin{bmatrix} C_1 & 0 \\ 0 & C_2 \end{bmatrix} \begin{bmatrix} \frac{dT_1}{dt} \\ \frac{dT_2}{dt} \end{bmatrix} + \begin{bmatrix} \frac{1}{R_1} & \frac{1}{R_1} \\ -\frac{1}{R_1} & \left(\frac{1}{R_1} + \frac{1}{R_2}\right) \end{bmatrix} \begin{bmatrix} T_1 \\ T_2 \end{bmatrix} = \begin{bmatrix} q_o \\ \frac{1}{R_2} T_o \end{bmatrix} \quad (25)$$

Let

$$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} T_1 \\ T_2 \end{bmatrix}, u = [q_o], y = [T_1]$$

Then, after little simplification to (25), one gets

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} -\frac{1}{R_1 C_1} & \frac{1}{R_1 C_1} \\ \frac{1}{R_1 C_2} & -\left(\frac{1}{R_1 C_1} + \frac{1}{R_2 C_2}\right) \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} \frac{1}{C_1} \\ 0 \end{bmatrix} u \quad (26)$$

$$y = [1 \quad 0] \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + [0] u$$

Which is a SISO linear state-space model for the room heater depicted in Fig.4. It is worthy to remember that the state vector x_1 and x_2 should not be confused with input of the ENN $x_i(k)$.

6. SIMULATIONS RESULTS

A thermal process linear system of Fig. 4 described by the (25) is considered for the purpose of the simulations. The values of thermal capacities are $C_1 = C_2 = 1/6$, while thermal resistances are $R_1 = R_2 = 2.5$. Based on the above, the transfer fuction of the linear system is

$$\frac{Y(s)}{X(s)} = \frac{14.4}{s^2 + 7.2s + 5.76}$$

Discretizing the above transfer function with zero-order holder and sampling interval T_s of 0.1, one gets the discretized model,

$$\frac{Y(z)}{X(z)} = \frac{0.05717 Z + 0.04499}{Z^2 - 1.446 Z + 0.4868}$$

Upon rearranging the two sides of the above equation, the difference equation that governs the thermal process system is given as

$$Y(k) = 1.446Y(k-1) - 0.4868 Y(k-2) + 0.05717x(k-1) + 0.04499 x(k-2)$$

Which is of the form

$$Y(k) = f \{ Y(k), Y(k-1), Y(k-2), x(k-1), x(k-2) \}$$

Our aim is to train the ERNN with a random i/p $x(k)$ in the range of $[-2,+2]$ so that the o/p of the ENN is similar to that of the dynamic thermal process system for the same i/p. Selecting a suitable length of 300 for the random i/p vector within the range of $[-2,+2]$ and assuming zero initial conditions for the i/p and o/p, i.e. $Y(-1) = Y(-2) = 0$ and $x(-1) = x(-2) = 0$, one can train ENN with the following ENN data: a linear activation functions at the i/p and o/p layers of the ERNN, log-sigmoid activation function in the hidden layer with linear ones has been used in the i/p and o/p layers, whereas the linear activation function acts as a buffers and as a summer at the o/p layers, weights updating algorithm (trainlm) is Levenberg-Marquardt optimization, random initialization for all of the weights, maximum number of iterations is set to 700, performance goal = 0.0005, number of hidden layer neurons is taken as 25, 35. e The ERNN have been trained hoping that it will produce the requested response when the i/p data is imposed on it. The random i/p vector is applied to both the plant dynamics of the thermal process and the trained ERNN and the results are drawn in Fig. 5. The ERNN outperform the traditional CFFNN regarding the smoothness and the accuracy of the identification process. In Fig.6, the o/p's of the trained ENN, CFFNN, and thermal plant process are depicted with 25 neurons in the hidden layer for ERNN and CFFNN.

By generating a valid and deterministic i/p data to test the trained ERNN, the following mathematical nonlinear functions has been programmed and simulated in MATLAB to produce the realistic input vector necessary for testing the trained ERNN and CFFNN.

1. $f(k) = \begin{cases} \cos(0.02\pi k) & 0 \leq k \leq 150 \\ \sin(0.1\pi k) & 150 < k \leq 300 \end{cases}$
2. $g(k) = \sin(0.1\pi k) * \exp(-0.01k), \quad 0 < k \leq 300$
3. $h(k) = \begin{cases} 2 & 0 \leq k \leq 150 \\ -2 & 150 < k \leq 300 \end{cases}$

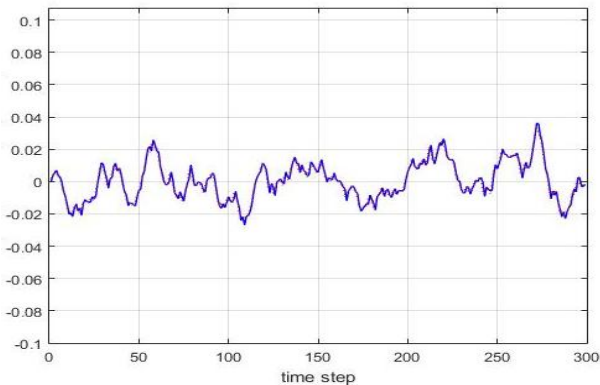
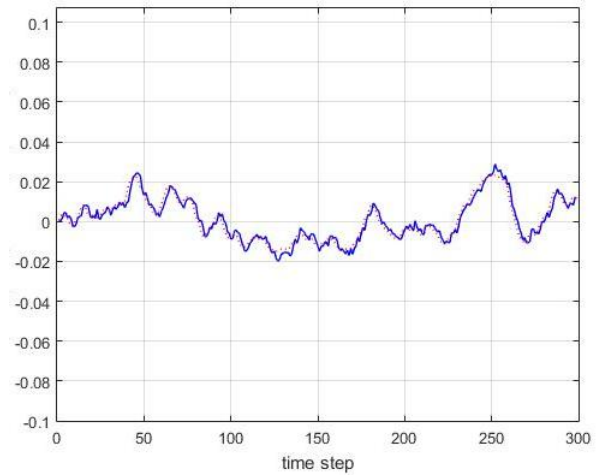
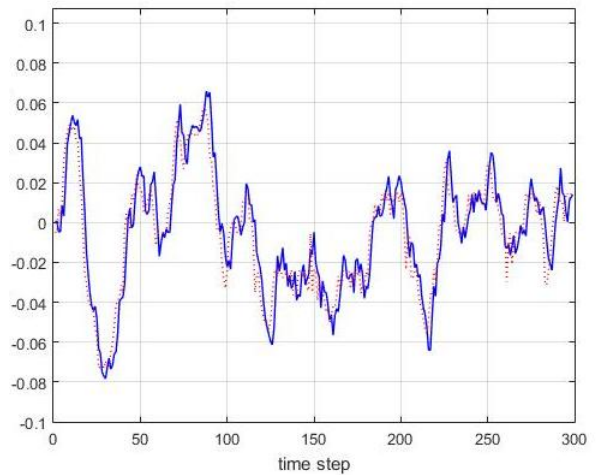


Fig 5:Trained ERNN and Thermal Model o/p's for the same random i/p vector with 35 neurons in the hidden layer of the ERNN, solid line (blue) is the plant o/p and dotted line (red) is the ERNN o/p.

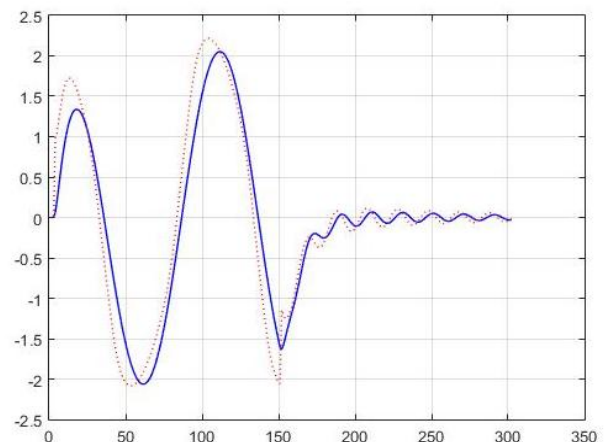


(a)

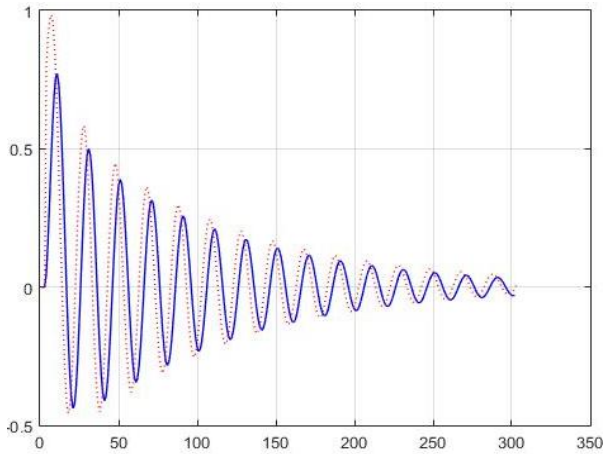


(b)

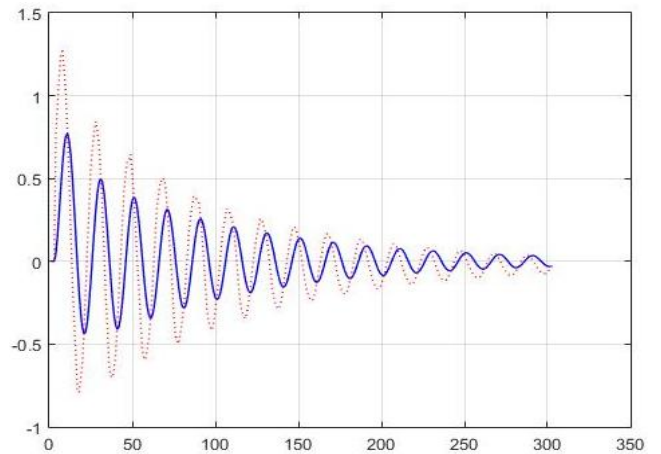
Fig 6: Trained ENN and Thermal Model o/p's for the same random i/p vector with 25 neurons in the hidden layer for both ERNN and CFFNN, (a) solid line (blue) is the plant o/p and dotted line (red) is the ERNN o/p, (b) solid line (blue) is the plant o/p and dotted line (red) is the CFFNN o/p.



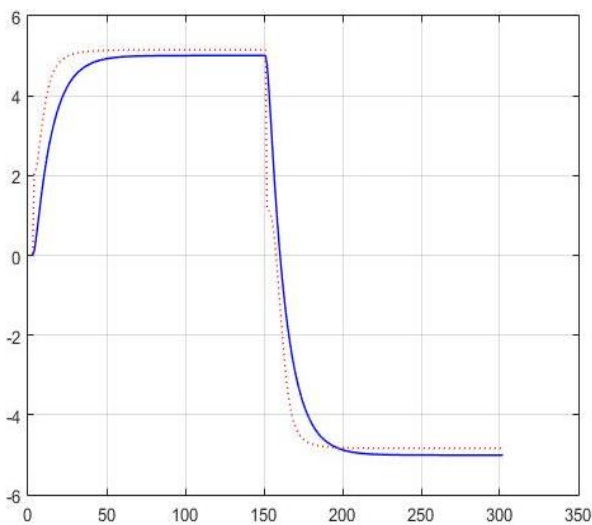
(a)



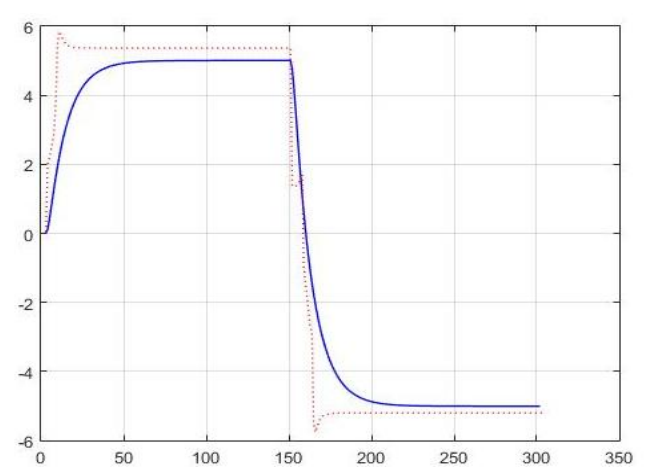
(b)



(b)



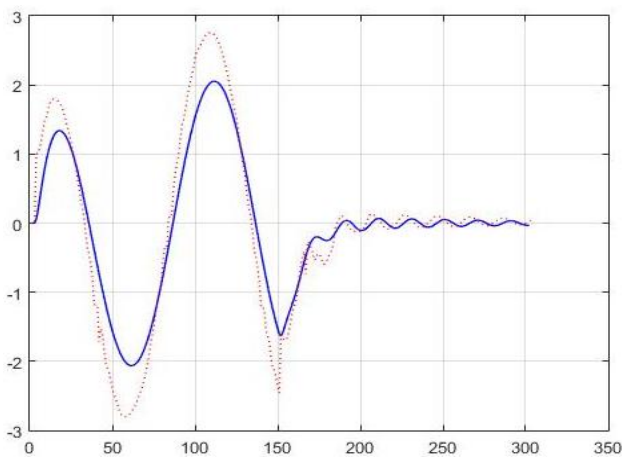
(c)



(c)

Fig 7: Applying benchmark functions to the trained ERNN and thermal plant, solid line (blue) is the plant o/p and dotted line (red) is the ENN o/p, (a) plant and ENN o/ps for $f(k)$, (b) plant and ENN o/ps for $g(k)$, (c) plant and ENN o/ps for $h(k)$.

Fig 8: Applying benchmark functions to the trained conventional Feedforward Neural Networks (CFFNN) and thermal plant, solid line (blue) is the plant o/p and dotted line (red) is the FFNN o/p, (a) plant and FFNN o/ps for $f(k)$, (b) plant and FFNN o/ps for $g(k)$, (c) plant and FFNN o/ps for $h(k)$.



(a)

The results of applying the above benchmark functions for the trained ERNN and CFFNN are illustrated in Fig.7 and Fig. 8. As can be seen from these Figures, the ERNN produces a smooth o/p, nearly coincident on the thermal plant o/p. The reason is that ERNN better remembers the previous states of the system due to the context layer which works as a short memory that stores the last states of the thermal plant dynamics. It is worthy to mention that there are no sharp changes in the o/p responses of the ERNN as compared the o/p response of the CFFNN in Fig.7.

7. CONCLUSIONS

In this paper, system identification method based on ERNN has been proposed for thermal process assuming no prior knowledge of system dynamics, and another identification method based on CFFNN for the same plant model has been obtained. A black-box model has been adopted for both neural networks models. With three nonlinear test functions plus a random i/p vector used to check the correctness of the trained models, it can be verified that the identified model using ERNN responds more smoothly and accurately than CFFNN to these test functions even with some discontinuities in the i/p test functions. Through simulations and results

demonstrations, ERNN produces better results due to the context layer which acts as memory stage to remember the last states of the system dynamics. The proposed method can be applied to different applications as well. The future work for this research paper is to handle the nonlinearity in the modeling of the thermal process system and applying a new trend of neural networks called convolutional neural network (CNN) is a type of feed-forward artificial neural network in which the connectivity pattern between its neurons is inspired by the organization of the animal visual cortex.

8. REFERENCES

- [1] Dong, Ze, Pu Han, Dongfeng Wang, and Songming Jiao. "Thermal Process System Identification Using Particle Swarm Optimization." In *2006 IEEE International Symposium on Industrial Electronics*, vol. 1, pp. 194-198. IEEE, 2006.
- [2] Liu, Tao, Ke Yao, and Furong Gao. "Identification and autotuning of temperature-control system with application to injection molding." *IEEE transactions on control systems technology* 17, no. 6 (2009): 1282-1294.
- [3] Esfandiari, Ramin S., and Bei Lu. *Modeling and analysis of dynamic systems*. CRC Press, 2014.
- [4] Linhares, Leandro LS, Aluisio IR Fontes, Allan M. Martins, Fábio MU Araújo, and Luiz FQ Silveira. "Fuzzy Wavelet Neural Network Using a Correntropy Criterion for Nonlinear System Identification." *Mathematical Problems in Engineering* 2015 (2015).
- [5] Toha, Siti Fauziah, and M. Osman Tokhi. "MLP and Elman recurrent neural network modeling for the TRMS." In *Cybernetic Intelligent Systems, 2008. CIS 2008. 7th IEEE International Conference on*, pp. 1-6. IEEE, 2008.
- [6] Rout, Ajit Kumar, P. K. Dash, Rajashree Dash, and Ranjeeta Bisoi. "Forecasting financial time series using a low complexity recurrent neural network and evolutionary learning approach." *Journal of King Saud University-Computer and Information Sciences* (2015).
- [7] Horvath, Gábor. "Neural networks in system identification." *Nato Science Sub-Series III Computer And Systems Sciences* 185 (2003): 43-78.
- [8] Scarpiniti, Michele, Danilo Comminiello, Raffaele Parisi, and Aurelio Uncini. "Novel cascade spline architectures for the identification of nonlinear systems." *IEEE Transactions on Circuits and Systems I: Regular Papers* 62, no. 7 (2015): 1825-1835.
- [9] Liu, Changliang, and Xiaojiao Sun. "Electromagnetism-like mechanism particle swarm optimization and application in thermal process model identification." In *Control and Decision Conference (CCDC), 2010 Chinese*, pp. 2966-2970. IEEE, 2010.
- [10] Maachou, Asma, Rachid Malti, Pierre Melchior, Jean-Luc Battaglia, Alain Oustaloup, and Bruno Hay. "Nonlinear thermal system identification using fractional Volterra series." *Control Engineering Practice* 29 (2014): 50-60.
- [11] Pan, Wei, Ye Yuan, Jorge Gonçalves, and Guy-Bart Stan. "A sparse Bayesian approach to the identification of nonlinear state-space systems." *IEEE Transactions on Automatic Control* 61, no. 1 (2016): 182-187.
- [12] Scarpiniti, Michele, Danilo Comminiello, Raffaele Parisi, and Aurelio Uncini. "Nonlinear system identification using IIR spline adaptive filters." *Signal Processing* 108 (2015): 30-35.
- [13] Tavakolpour-Saleh, A. R., S. A. R. Nasib, A. Sepasyan, and S. M. Hashemi. "Parametric and nonparametric system identification of an experimental turbojet engine." *Aerospace Science and Technology* 43 (2015): 21-29.
- [14] Li, Zongjian, Yuling Chen, Weichao Fang, Zhengjiang Zhang, Guoqiang Zeng, and Yuxing Dai. "Research on parameter identification method based on finite measurement information for photovoltaic array model." In *Control Conference (CCC), 2014 33rd Chinese*, pp. 6465-6470. IEEE, 2014.
- [15] Shobana, R., C. Sreepadha, S. Sobana, and Rames C. Panda. "Identification and Estimation of temperature in Nitration process." In *Technological Innovation in ICT for Agriculture and Rural Development (TIAR), 2015 IEEE*, pp. 165-170. IEEE, 2015.
- [16] Wang, Jeen-Shing, and Yen-Ping Chen. "A fully automated recurrent neural network for unknown dynamic system identification and control." *IEEE Transactions on Circuits and Systems I: Regular Papers* 53, no. 6 (2006): 1363-1372.
- [17] Sebakhy, O. A., H. M. A. Kader, W. A. Youssef, and S. Deghiedi. "Identification of linear discrete time systems using linear recurrent neural networks." In *Industrial Electronics, 1996. ISIE'96., Proceedings of the IEEE International Symposium on*, vol. 1, pp. 374-379. IEEE, 1996.
- [18] Nouri, Khaled, Rached Dhaouadi, and N. Benhadj Braiek. "Identification of a nonlinear dynamic systems using recurrent multilayer neural networks." In *Systems, Man and Cybernetics, 2002 IEEE International Conference on*, vol. 5, pp. 5-pp. IEEE, 2002.
- [19] Fan, Bo, Xing Li, Guanghui Shi, and Weigang Zhao. "Motor rotor resistance identification based on Elman neural network." In *2012 IEEE International Conference on Automation and Logistics*, pp. 196-200. IEEE, 2012.
- [20] Zhang, Wenjun, Zhengjiang Liu, Jinshan Zhu, and Xiaoka Xu. "Identification and control of time-delay system with recurrent wavelet neural networks." In *Intelligent Control and Information Processing (ICICIP), 2012 Third International Conference on*, pp. 211-216. IEEE, 2012.
- [21] Khalil, Rafid Ahmed. "Comparison of Four Neural Network Learning Methods Based on Genetic Algorithm for Non-linear Dynamic Systems Identification." *Al-Rafadain Engineering Journal* 20, no. 1 (2012).
- [22] array, Fakhreddine O., and Clarence W. De Silva. *Soft computing and intelligent systems design: theory, tools, and applications*. Pearson Education, 2004.
- [23] Nguyen, Hung T., Nadipuram R. Prasad, Carol L. Walker, and Elbert A. Walker. *A first course in fuzzy and neural control*. CRC press, 2002.
- [24] Narendra, Kumpati S., and Kannan Parthasarathy. "Identification and control of dynamical systems using neural networks." *IEEE Transactions on neural networks* 1, no. 1 (1990): 4-27.

- [25] Gao, X. Z., X. M. Gao, and S. J. Ovaska. "A modified Elman neural network model with application to dynamical systems identification." In *Systems, Man, and Cybernetics, 1996., IEEE International Conference on*, vol. 2, pp. 1376-1381. IEEE, 1996.
- [26] Kalinli, Adem, and Seref Sagiroglu. "Elman Network with Embedded Memory for System Identification" *Journal of Information Science and Engineering* 22 (2006): 1555-1568.
- [27] Pham, D. T., and X. Liu. "Training of Elman networks and dynamic system modelling." *International Journal of Systems Science* 27, no. 2 (1996): 221-226.
- [28] Ogata, Katanhiko. "Modern Control Engineering,(1997)." *ISBN: 0-13-227307-1*: 299-231.