Application of Artificial Neural Networks for the Modeling and Control of Nonlinear Systems

DISSERTATION

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE AWARD OF THE DEGREE OF

MASTER OF TECHNOLOGY
IN
CONTROL AND INSTRUMENTATION

Submitted by:

PRASHASTI SRIVASTAVA
Roll No. 2K20/C&I/06

Under the supervision
Of
DR. RAJESH KUMAR

DEPARTMENT OF ELECTRICAL ENGINEERING
DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Bawana Road, Delhi-110042

2022

DEPARTMENT OF ELECTRICAL ENGINEERING

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Bawana Road, Delhi-110042

CANDIDATE'S DECLARATION

I, Prashasti Srivastava, Roll No. 2K20/C&I/06, student of M.Tech (Control & Instrumentation), hereby declare that the thesis titled "APPLICATION OF ARTIFICIAL NEURAL NETWORKS FOR THE MODELING AND CONTROL OF NONLINEAR SYSTEMS" which is submitted by me to the Department of Electrical Engineering, Delhi Technological University, Delhi in partial fulfillment of the requirement for the award of the degree of Master of Technology, is original and not copied from any source without proper citation. This work has not previously formed the basis for the award of any Degree, Diploma Associateship, Fellowship or other similar title or recognition.

Place: Delhi **PRASHASTI SRIVASTAVA**

Date:

DEPARTMENT OF ELECTRICAL ENGINEERING

DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Bawana Road, Delhi-110042

CERTIFICATE

I hereby certify that the thesis titled "APPLICATION OF ARTIFICIAL NEURAL NETWORKS FOR THE MODELING AND CONTROL OF NONLINEAR SYSTEMS" which is submitted by Prashasti Srivastava, 2K20/C&I/06 [Electrical Engineering Department], Delhi Technological University, Delhi in partial fulfillment of the requirement for the award of the degree of Master of Technology, is a record of the project work carried out by the student under my supervision. To the best of my knowledge this work has not been submitted in part or full for any Degree or Diploma to this University or elsewhere.

Place: Delhi
Date:

Dr. Rajesh Kumar
(Supervisor)
Department of Electrical Engineering, DTU

ACKNOWLEDGEMENT

I would like to start by thanking my supervisor, Dr. Rajesh Kumar. Without whom, I would not have made it through the research work. I am highly indebted to his constant guidance and unwavering support on all the matters pertaining to the work. I would also like to express my gratitude towards all the faculty members of the Department of Electrical Engineering along with the authorities of Delhi Technological University for their kind cooperation and encouragement in completion of my thesis. Lastly, I would like to thank my parents, family members and friends for always motivating me to strive high and bringing out the best in me.

ABSTRACT

In this thesis, the wide area of recurrent neural networks is explored to propose a novel structure for the purpose of prediction and control of nonlinear systems with unknown dynamics. The proposed structure is a locally recurrent neural network with input feed through (LRNNIFT) which consists of locally recurrent loops along with input fed through weights directly to the output. The proposed network model parameters are tuned using a Back-propagation (BP) algorithm. The performance of the proposed model is compared with the state-of-the-art recurrent Elman neural network (ENN) and a single layer feed-forward neural network (FFNN). The simulation results showed that the proposed model has shown better accuracy as compared to the other two models. Furthermore, the above network is presented for control of nonlinear dynamical systems. The rationale of using LRNNIFT is due to its modest structure and proven superiority in mathematical modeling. Results from simulation showed that the LRNNIFT based controller is able to achieve adaptive control in a nonlinear system. It is also tested and observed to counterbalance the effects of disturbances. A comparative analysis is presented with the help of simulation, and it is deduced that overall performance of the LRNNIFT controller is better than that of FFNN and ENN controllers.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

ANN	Artificial neural networks
FFNN	Feed forward neural networks
SLFFNN	Single layer feed forward neural networks
MLFFNN	Multi layer feed forward neural networks
RNN	Recurrent neural networks
LRNN	Locally recurrent neural networks
LRNNIFT	Locally recurrent neural networks with input feed through
BP	Back Propagation
IF	Induced Field
MSE	Mean square error
MAE	Mean average error
TMSE	Total Mean square error
TMAE	Total Mean average error

CHAPTER 1

INTRODUCTION

1.1 Thesis Organization

The first chapter is an introduction to set the premise: basic principles of soft computing, overview and advantage of ANNs, various neural network topologies used and the objective of work. Chapter 2 presents an exhaustive literature survey. This is followed by Chapter 3 which describes the proposed structure and its mathematical modelling. Chapter 4 explains a nonlinear system supported by equations and the identification scheme using the proposed structure. It also describes the learning algorithm along with the mathematical equations for weight adjustment. Chapter 5 discusses the simulation results of the structure as an identifier obtained by taking a system for analysis. Chapter 6 is the second part of the work, it describes the model of the previously proposed structure as a controller along with the layout of the control scheme. Finally, in Chapter 7, simulation results obtained by implementing the controller on another complex nonlinear system is presented with a disturbance rejection test to prove its adaptive nature.

1.2 Soft Computing Techniques

The world around us is imprecise, uncertain, and randomly changing. However, we can cope with such an environment. The desire to mimic such coping leads to the basic premises and the guiding principles of soft computing.

The three most popular soft computing methods are as below:

- Genetic algorithms
- Fuzzy logic
- Artificial Neural Networks

Genetic Algorithms are used to solve problems based on principles of natural selection that come under evolutionary algorithms. They are usually used for optimization problems like maximization and minimization of objective functions, which are of two types of an ant colony and swarm particle. It follows biological processes like genetics and evolution. The fuzzy logic algorithm is used to solve the models which are based on logical reasoning like imprecise and vague. It was introduced by Latzi A. Zadeh in 1965. Fuzzy logic provides a stipulated truth value with the closed interval [0,1]. Where 0 = false value, 1= true value. In the following subsection ANN is described in detail.

1.3 Overview on ANN

Artificial neural networks (ANN), as the name implies, the baseline for their development was the realization that the human brain does not work like a computer at all. It has a superior sense for intangibility and approximation. The brain can be treated as a highly complex computer that can process extremely large amount of data. The brain is a part of the nervous system where sensory organs provide the brain with all the data of human experiences. This transmission is done by an element known as neurons or nodes. They perform specific calculations (e.g., pattern recognition, vision, and motor control) and provides appropriate output we need to engage with nature. This makes the human as the most intelligent creature on earth. Specifically, the brain typically performs visual tasks at a rate much higher than that of a very powerful computer. A neural network is an algorithm designed to model how the brain performs a specific task or activity of interest; The process used to train such networks is called a learning algorithm, whose function is to change the tune the adjustable weights of a network in a systematic way to achieve the desired goal. Here, the backpropagation algorithm is used.

In a nutshell, an artificial neural network or ANN is an algorithm loosely based on the working of the human brain. Just as the brain consists of thousands of neurons and synapses associated with them, an artificial neural network also contains a set of nodes (known as neurons) that are interconnected via weights. In a sense, the weights determine the degree to which the information is to be transmitted to another node. Hence, they play a vital role in the working of a neural network. These weights are updated during the training process using an error reduction algorithm to produce desired output. Neural networks are used as black box models as they require only the input and output data set to draw patterns and to estimate the mathematical relationship between the output and input of a system. On a broad structural classification, artificial neural networks can be of two sorts: the feed forward neural network (FFNN) and the recurrent (or repetitive) neural network (RNN). Both of these networks contain characteristics which makes them an ideal candidate to use flexible real-time dynamic controls.

1.4 Advantages of ANN

I. Nonlinearity

Neural network is designed in a way where the connections between neurons are themselves nonlinear in nature. Additionally, this inherent nonlinearity is distributed across the network which becomes a major advantage. This makes these networks functionally the most superior to handle or process nonlinear inputs. It provides faster and accurate computation.

II. Input–Output Mapping

There are broadly three types of learning methodologies to train such networks. They are supervised learning, unsupervised learning and reinforcement learning. In supervised learning, an algorithm governs the network as a teacher or supervisor. Here the input provides an output based on some calculation within the network. For training the network, difference between the actual and desired output is calculated which acts as an error signal on the basis of which appropriate tuning of weights happen. This process is done until the error falls below a permissible threshold. This gives us the adjusted weights that can be used for application of this network. This exhibits the input output mapping nature of a neural network. Just a set of input output data can help predict and map a neual network further outputs for an input. This type of training is implemented in our analysis as well. Supervised learning is unlike the algorithm stated previously. The network is independent of any supervision algorithm. Here, input vectors of the same type are combined to form clusters of data. If a new input pattern is introduced, then the neural network provides an output response that shows the input pattern class. Therefore, in this type of learning, the network itself should find patterns and features in the input data, as well as the correlation of the input data over the output. Reinforcement Learning, as the name suggests, this type of learning is used to strengthen or reinforce the network with some critical information. This learning process is similar to supervised learning, yet we may have very little knowledge. During network training under reinforced learning, the network receives feedback from the site. This makes it look like a supervised reading. However, the answer found here is non-conclusive but rather evaluative, which means that no teacher is as supervised as reading. After receiving a response, the network underwent weight changes to obtain better criticism information in the future.

III. Adaptivity

Neural networks have a built-in ability to adapt and adjust their weights to acclimate with the changes in the system environment. Particularly, a neural network trained to work specifically on a solution can be smoothly re-trained to deal with small disturbances in environmental conditions. In addition, when operating in a stable environment, a neural network can be designed to change its synaptic weights in real time by implementing online training. The natural structure of the neural network of pattern classification, signal processing, and application controls, combined with the network's adaptive capabilities, make it a useful tool for dynamic pattern detection, dynamic signal processing, and dynamic control. As a general rule, it can be said that when we make a flexible system, we always ensure that the system remains stable, its performance will be even stronger when the system is required to operate in a stable environment. To check the full benefits of adaptability from training standpoint, the main fixed times of a program should be long enough so that the system ignores the false positives, but be short enough to respond to actual changes in the system.

IV. Fault Tolerance

A neural network when implemented in hardware such as power systems, have the ability to tolerate faults. For example, in case if there is a neuron or its connecting weight that is compromised, memory retention deteriorates in quality. But, due to the distributed nature of the information stored on the network, the damage must be major enough for the total network response to be significantly altered. Thus, from experimentation, the neural network shows a better deterioration in function than catastrophic failure. This very attribute of a neural network is further displayed in this work as a part of robustness analysis.

1.5 Network Topologies

In general, we may identify two fundamentally different classes of network architectures as described in the upcoming subsections.

1.5.1 Feed forward networks (FFNN)

The first type of neural network is the feed forward neural networks. They are simple yet robust networks where the signal flows from the input to the output only, hence unidirectional in nature. These networks can be of two types- single layer feed forward networks (SLFFNN) and multilayer feed forward neural networks (MLFFNN).

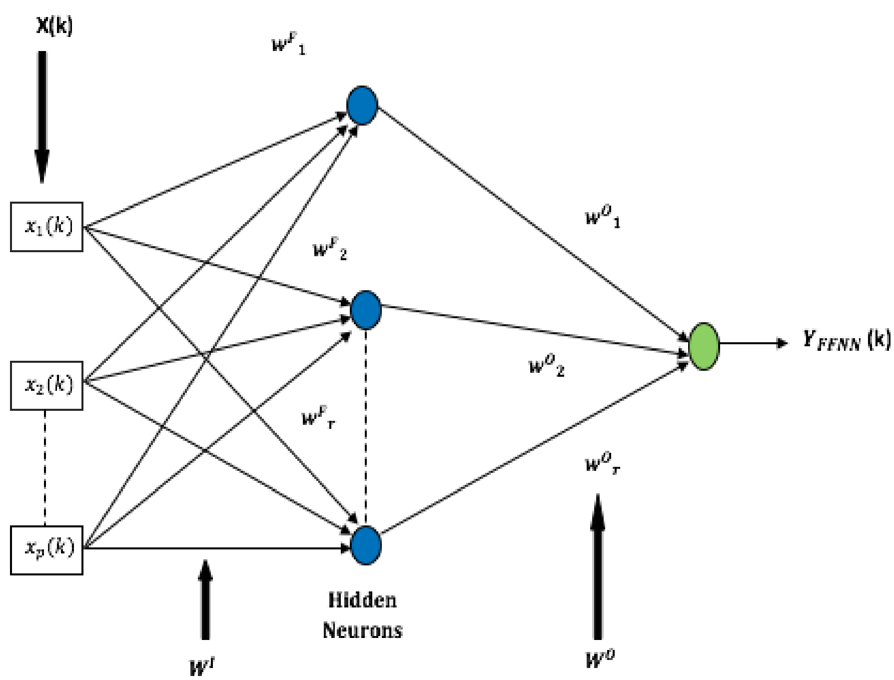


Fig 1: Single layer feed forward neural network (SLFFNN)

These networks are differentiated by the presence of one or more hidden layers for SLFFNN and MLFFNN respectively. The term "hidden" refers to the fact that this part of the neural network is not determined directly from the input or output of the network.

The reason why these hidden layers are added to the network, even though they increase the computational complexity is because it helps generate more useful information for the network in a faster and accurate manner. Meaning, the addition of extra parameters (weights due to hidden layer) increases the efficiency of the network. This modification is necessary if one is dealing with highly complex network, in our case a nonlinear plant whose dynamics aren't known will be such system. Each node of the network contains an activation function based on computational requirements. In the figure above, we can observe a simple 3-layer (input, hidden and output) SLFFNN. This network is used as a one of the bases for comparison in our analysis as is it one of the most widely used, simple yet efficient networks. Moving forward, for the sake of simplicity this circuit will be called as FFNN only.

1.5.2 Recurrent Neural networks

A neural network is said to be recurrent in nature if it has loops in it. A connection where the output of a particular neuron goes back to a previous or the same neuron as an input is called a loop. This means the there is a bidirectional movement of signal in such networks, unlike FFNNs.

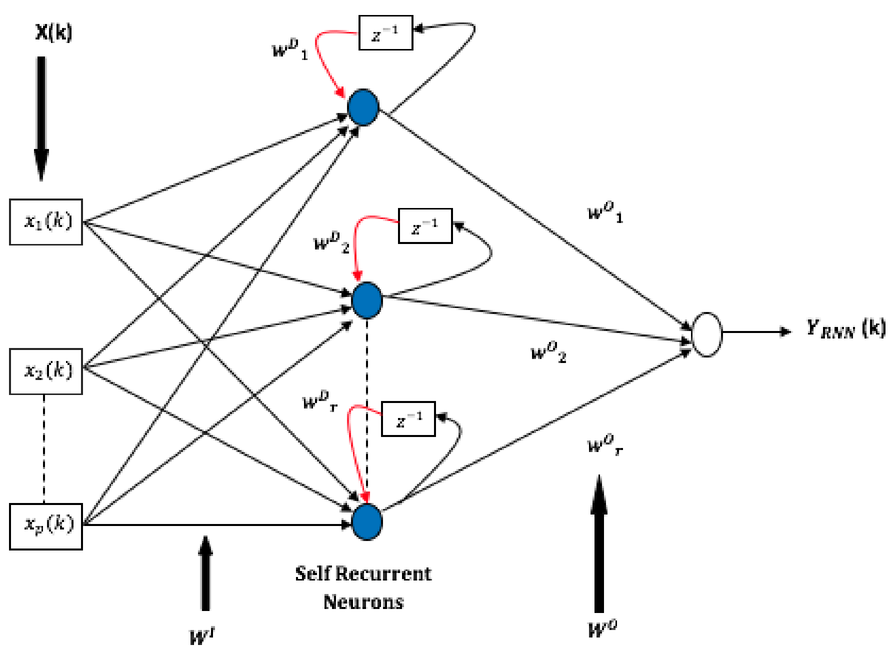


Fig 2: Recurrent neural network (RNN)

The presence of such loops indicates a feedback mechanism in the network and when the signal in the loop is propagated via a delay, it develops a certain kind of memory. This attributes to the heavy nonlinearity of such networks. In our context, we can say that the circuit somehow retains the past values of the system to predict the present values. Therefore, these networks are widely used for the purpose of prediction. Figure 2 represents a recurrent neural network which has time delayed local recurrent loops within the hidden layer. This network is further modified to propose a novel structure.

The below figure (Fig 3) represents the structure of an elman neural network (a type of RNN) which is used in this work as a basis of comparison as it is one of the most complex and popular recurrent neural networks.

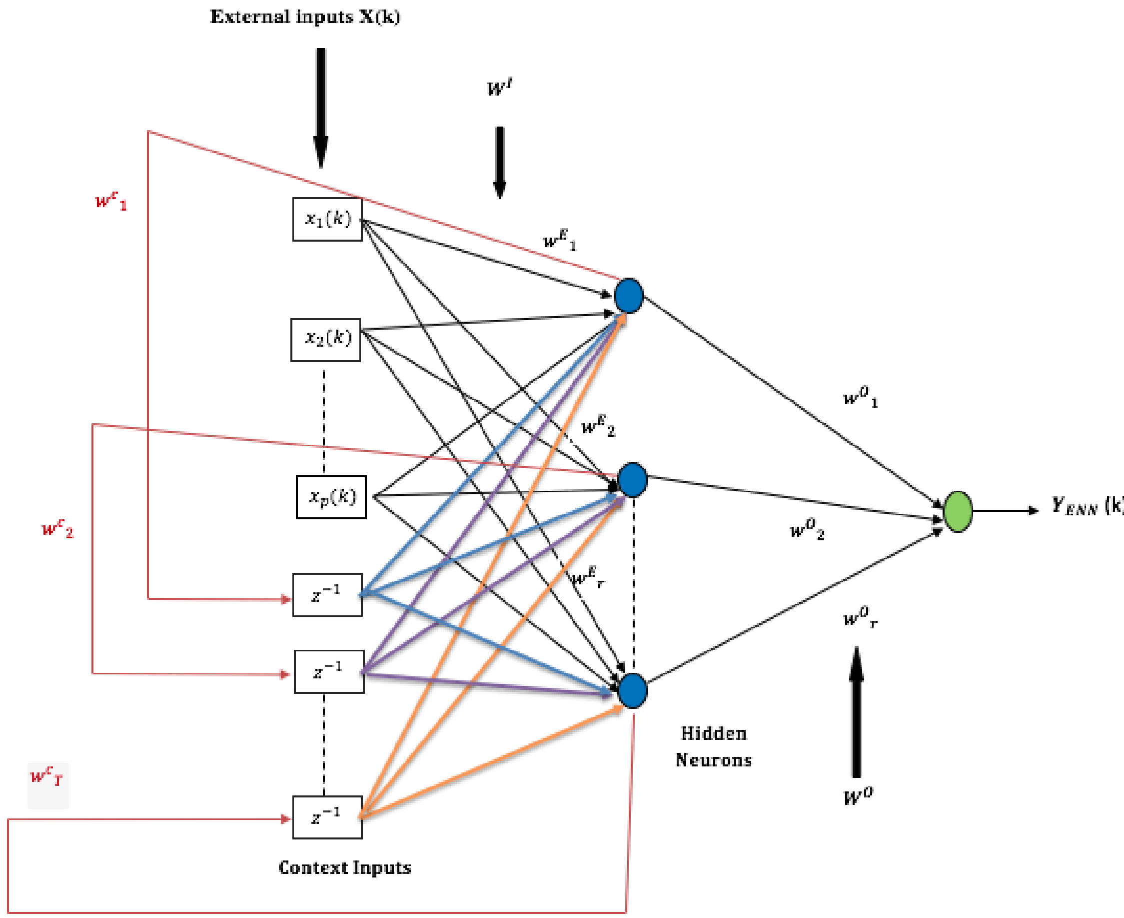


Fig 3: Elman Neural Network (ENN)

1.6 Objectives

1. To study about various Artificial Neural Networks and some of its structures and applying them for predicting models

2. To propose a better performing recurrent neural network structure to predict the behavior of nonlinear dynamical systems and compare the results with existing feed forward and elman neural networks

3. To present a detailed simulation of the LRNNIFT for prediction of a nonlinear time-delayed plant using backpropagation algorithm.

4. To compare the proposed identifier with identifiers based on some of the existing neural network structures such as feed forward neural network (FFNN) and Elman neural network (ENN)

5. The proposed neural network is further modeled as a controller for nonlinear plants and the results are compared with FFNN and ENN.

6. To test the proposed controller for robustness by introducing a disturbance signal

CHAPTER 2

LITERATURE REVIEW

In the ever-growing world of science, extensive research is being performed in understanding the behavior of the universe and its systems. The significance of nonlinear system identification is unarguable as most control methods are designed to perform in the linear region only and have a restricted range of working. It is evident that linear control methods are popular because they are simple but their effectiveness in real-world (nonlinear) system control is very limited. The linear identification process involves the selection of a specific appropriate phase of the model in order to best measure the behavior of a system. In [1], the authors have proposed a data driven bias eliminated subspace identification approach by using coprime factorization to approximate closed loop systems whereas in [2] the authors undertook a probabilistic approach to solve the problem of identification along with developing conventional error reduction functions such as in [3]. [4] encompasses an overview of a wide range of modern data driven identification approaches developed.

To overcome this problem, researchers started focusing on modeling advanced techniques for systems with partially known dynamics. Nonlinear system modeling techniques are versatile in nature and can work in a vast range unlike linear identification methods. In [5], authors have implemented a single layer feed forward neural network for an unsupervised learning algorithm based upon a Hebbian learning rule which helps in optimal optimisation. A single layer neural network being simple fails to converge for systems with higher complexity. This drives the implementation of multilayer feed forward networks as described in [6]. In [7], the Chebyshev polynomial based method is proposed to unify these two kinds of feed forward networks. Although better than a single layer neural network, a multilayer feed forward network could still successfully approximate such systems with desirable speed. Advancing to further modification of neural network structures, in [8] authors proposed the design and application of recurrent neural networks.

These networks had recurrent loops within the structure with allowed them to have a memory which in turn provided faster and accurate results. Elman neural network is a type of recurrent neural network that has widespread application in the field of approximation and control. [9] describes the use of ENN for identification of nonlinear systems whereas [10] encompasses it for short-term forecasting of wind power generation. A hybrid of quantised genetic algorithm and ENN is also proposed in [11]. Apart of neural networks, other biologically inspired algorithms such as particle swarm optimization [12] [13] and fuzzy logic [14] have also been implemented majorly for optimization purpose. This paved the way for the rise of the era of soft computing techniques, algorithms based on biological phenomena where a varied range of neural networks and optimisation algorithms were designed [15].

Modeling being the first part of the problem for such systems, is then followed by the second part, that is, control. One of the earliest significant control techniques in conventional control was the Ziegler Nichols method [16] which described rules to tune P, PI or PID controllers for linear systems. This was further followed by several modifications such as [17] describes a modified ZN method to tune fractional order systems and [18] presents a modification for multiple input and multiple output systems. Further modifications followed and advanced forced oscillation techniques [19][20] to achieve control on linear plants were also developed. Although [21] does present an auto tuning PID control using Runge-Kutta model solution for control of nonlinear systems, essentially all these methods were limited to linear systems only. Most processes that require control nowadays are nonlinear, the use of conventional proportional- integral-derivative controllers has seen a decrease in relevance as the majority of the plants are nonlinear in nature or whose dynamics are not fully known. Artificial neural networks [22][23] (ANN) are a major breakthrough [24] in this area as these structures help estimate as well as control nonlinear systems including industrial applications [25] such as control of a high-performance aircraft [26]. [27] presents a recurrent neural network for the identification as well as control of nonlinear systems using reinforcement learning. A hybrid of fuzzy logic control and ENN was presented in [28] to predict short-term power load. [29] describes an adaptive neural network with predictive control for multi-rate networked industrial control. Various structures of ANNs have been developed [30] over the time to cater to ever changing modern problems.

CHAPTER 3

Proposed Structure (LRNNIFT)

3.1 Description of the Structure

LRNNIFT is a locally recurrent neural network whose input is fed through weights to the output.

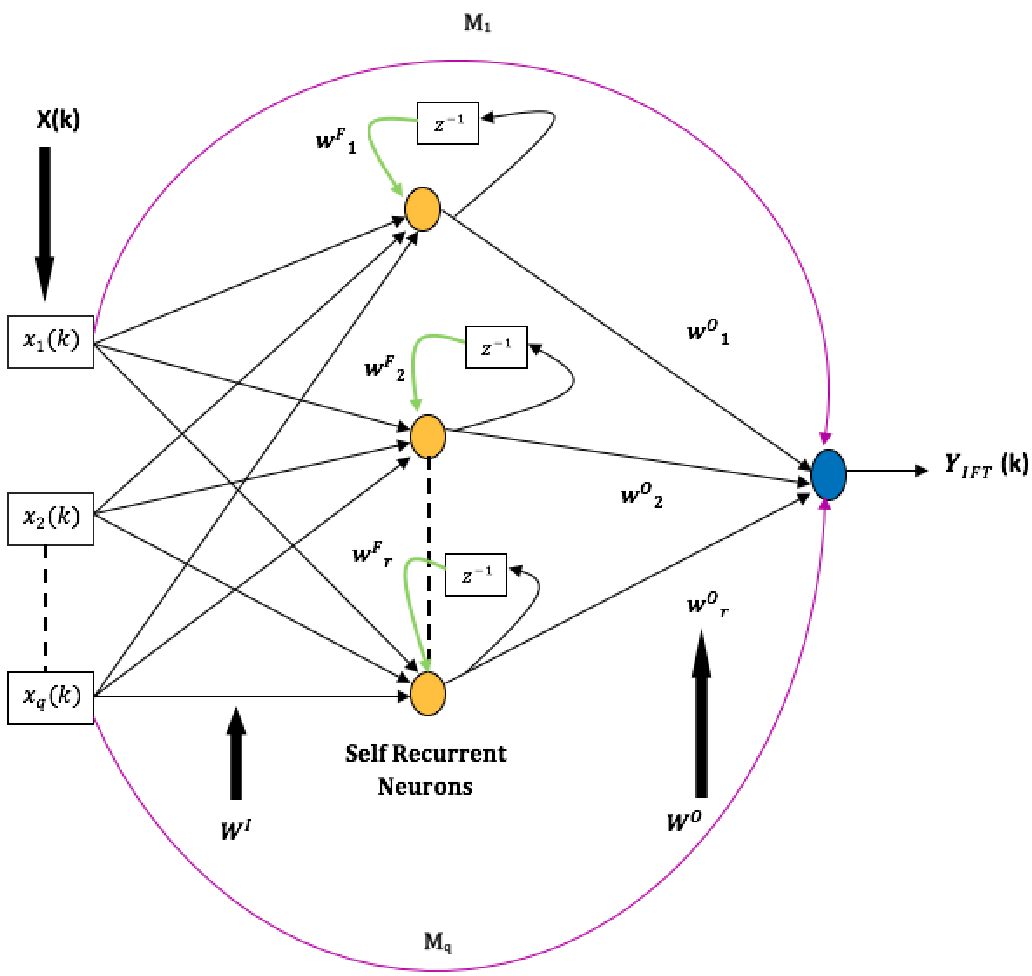


Fig 4: Structure of locally recurrent neural network with input feed through

In figure 4, the green arrows in the structure of LRNNIFT, depict the local weights which propagate a unit delay as output of the hidden neuron which is looped back as an input to the same hidden neuron. This forms the locally recurrent structure. The pink arrows denote the input fed to the output via feed-through weights represented as M= {M_1, M_2...M_q}. The local weights are given by the local weight vector, W_F = {w_1, w_2...w_q} and W_I represents the input weight vector having tunable weights. It can be deduced from the figure that if vector M_q is equal to zero, the structure of LRNNIFT reduces to an Elman Neural Network (ENN) and similarly if W^F is made zero, it reduces the structure to a Feed Forward Neural Network (FFNN). Both these structures are used further in this paper to compare the LRNNIFT structure. The output layer and hidden layer neurons are connected by a weight vector given as output weight vector, W_0 = {w_0^1, w_0^2 .... w_0^p} . The q-input vector depicts the external signal which is applied to the network and is represented as a vector by X= {x_1, x_2....x_q}.

3.2 Mathematical Formulation

Additionally, the output of any p^{th} hidden neuron at any k^{th} point in time is given by:

$$T_p(k) \; = \; f[Z_p(k)] \qquad (1)$$

Where f is a tangent hyperbolic function

The induced field of any p^{th} hidden neuron is calculated as below:

$$Z_p(k) = \; W_P^F(k) T_p(k-1) + \; \sum_q W_{pq}^I(\, k) x_q(k) \qquad (2)$$

Further, the induced field of the output neuron is as follows:

$$V(k) = \; \sum_p W_0^p \,(k) T_p(k) \qquad (3)$$

A linear activation function is considered for the output layer neuron; therefore, the output will be equal to its own induced field along with summation of external input via feed through weights. Subsequently, we can deduce:

$$V_{IFT}(k) = V(k) = \sum_p W_0^p(k) T_p(k) + \sum_q M_{pq}(k) x_q(k) \tag{4}$$

CHAPTER 4

Identification Scheme

4.1 Nonlinear Systems

A nonlinear plant can be of two types:

- Static Systems: Output depends on present values of input
- Dynamic Systems: Output of system depends on past values input & output

In our analysis, we have considered dynamic plants due to their high complexity.

4.2 Mathematical representation of a nonlinear dynamical system

In any control scheme, the basic feature is to provide the desired control of plant parameters. It is noteworthy to mention that in order to implement a control strategy on a plant, it is essential to know its dynamics. It is cumbersome to design a controller for plant which does not have a mathematical model. Subsequent to this, a major class of real time systems are present whose dynamics cannot be mathematically modeled and are non-deterministic in nature. This is where ANN has played a major role in recent times by helping estimate the unknown behaviour of nonlinear plants that are to be controlled.

The neural networks are backed by several robust learning algorithms which can be modified as per the nature of the system in consideration. With respect to this paper, neural networks are used as an identifier for a dynamical system whose properties are partially known. The general mathematical expression of a dynamical system is as follows:

$$y_q(k) = F[y_q(k-1), y_q(k-2) \ldots . y_q(k-O), r(k-1), r(k-2) \ldots r(k-L)] \qquad (5)$$

4.3 Identification scheme

As mentioned above, equation (5) represents a general dynamical system where $y_q(k-1)$ is the past value of plant (with a difference of one instant) and similarly all past values of $y_q(k)$output are represented till q^{th} instant. In the same manner, the past values of input are also represented as $r(k-1)$ to $r(k-L)$. The term O represents the order of the plant and L is generally taken to be less than O. It is to be noted that for the sake of simplicity in understanding, the above equation is represented for k^{th} time instant, that is, if the above equation is shifted by a unit delay, it will be visibly evident that the next value of the plant will depend upon the past as well as the present values of input and output. Therefore, in all further discussions this notion has been kept in mind. Further, the nonlinear system to be estimated is depicted as the function F.

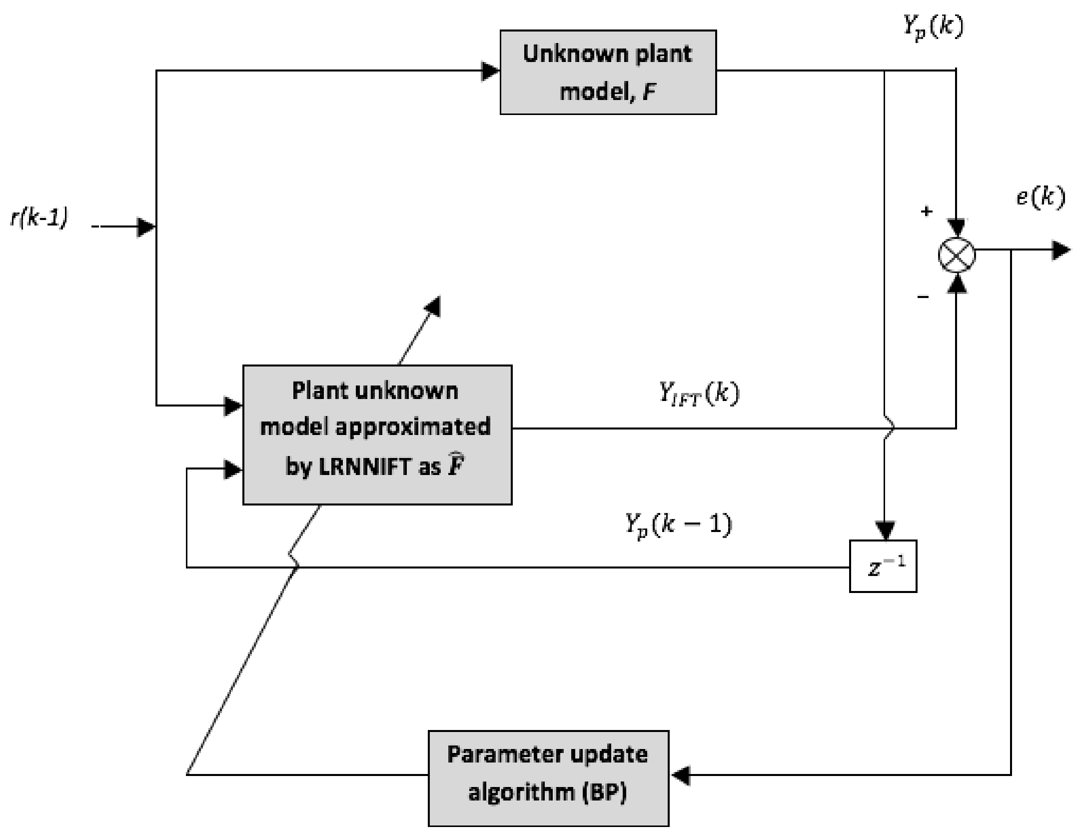


Fig 5: Model Identification scheme for LRNNIFT identifier

Our prime objective here is to estimate the behaviour of the above plant, that is, our end result should be $\hat{F} \approx F$. Now, from the structural layout we can conclude that the successive outputs of FFNN, Elman neural network and LRNNIFT structures are primarily dependent on the present and past values of the plant input-output values. The effectiveness of any estimation technique can be proven only if it minimizes the requirement of plant knowledge. Therefore, in case of LRNNIFT, the proposed identification structure, we have taken only two inputs from the large set of plant variables-one past value, $y_p(k-1)$and a past value of external input, $r(k-1)$. The plant behavior is predicted based on these two inputs only and is denoted by $Y_{IFT}(k)$. The selection of fewer inputs (in this case, two) tones the structural and computational complexity by reducing the number of parameters that are to be updated.

Fig. 5 depicts the block diagram of the proposed LRNNIFT model having below equation:

$$V_{IFT}(k) = F[y_p(k-1), r(k-1)] \tag{6}$$

4.4 Learning algorithm: Backpropagation Method

The Back propagation is a technique specific for implementing gradient descent in weight space for a multilayer neural network. The main idea behind the algorithm is to find the maximum or minimum (in this case, minimum) of a function. It uses the gradient at present value (by calculating the partial derivative) to iteratively compute the next value while minimising the function. Our aim is to generate an LRNNIFT based identifier which can estimate the plant's behaviour successfully. In order to attain this, we need to reduce the error (the gap between desired and actual output) generated during training. Conventionally, mean square error (MSE) function is used as the function to be minimised as it provides stable output compared to absolute error. The same is used below and expressed as:

$$E_i(k) = \frac{1}{2}[Y_p(k) - Y_{IFT}(k)]^2 = \frac{1}{2}e_i^2(k) \tag{7}$$

Here, $E_i(k)$ represents the cost function for the LRNNIFT training model and $e_i\ (k)$ represents the error which is back propagated. The input weights, locally recurrent weights, feed through weights and the output weights are to be updated in this model. The parameter update expression of all these are presented subsequently below.

4.5 Tuning of Weights

4.5.1 Tuning of output and feed-through weights

The error gradient with respect to any j th output layer weight can be expressed as

$$\frac{\partial E_i(k)}{\partial W_o^j(k)} = \left(\frac{\partial E_i(k)}{\partial Y_{IFT}(k)} \times \frac{\partial Y_{IFT}(k)}{\partial W_o^j(k)}\right) \quad (8)$$

Where $\frac{\partial Y_{IFT}(k)}{\partial W_o^j(k)} = T_j(k)$

Thus, each value in $W_0(k) = w_0^1(k), w_0^2(k) \ldots\ldots w_0^q(k)$ will be updated to a new value as following:

$$W_0^j(k) = W_0^j(k-1) - \triangle W_0^j(k-1) \quad (9)$$

Where $\triangle W_0^j(k-1) = -\eta e_i(k-1)T_j(k-1)$

In a similar manner for the feed through weights, each value in M= {M_1, M_2,...M_q} will be updated to a new value as following:

$$M_0^j(k) = M_0^j(k-1) - \triangle M_0^j(k-1) \quad (10)$$

where $\triangle M_0^j(k-1) = -\eta e_i(k-1)T_j(k-1)$

4.5.2 Tuning of locally recurrent weights

The update equation for any j th recurrent weight vector can be written as below:

$$\frac{\partial E_i(k)}{\partial W_F^j(k)} = \left(\frac{\partial E_i(k)}{\partial Y_{IFT}(k)} \times \frac{\partial Y_{IFT}(k)}{\partial W_F^j(k)}\right) \tag{11}$$

where $\frac{\partial Y_{IFT}(k)}{\partial W_F^j(k)} = \frac{\partial Y_{IFT}(k)}{\partial V(k)} \frac{\partial V(k)}{\partial W_F^j(k)}$

Now, $\frac{\partial Y_{IFT}(k)}{\partial V(k)} = 1$

Hence, evaluation of $\frac{\partial V(k)}{\partial W_F^j(k)}$ is as below:

$$\frac{\partial V(k)}{\partial W_F^j(k)} = \frac{\partial V(k)}{\partial T_j(k)} \frac{\partial T_j(k)}{\partial W_F^j(k)} \tag{12}$$

Where $\frac{\partial V(k)}{\partial T_j(k)} = W_0^j(k)$ and,

$$\frac{\partial T_j(k)}{\partial W_F^j(k)} = \left(1 - T_j^2(k)\right)[T_j(k-1) + W_F^j(k)L_j(k-1)] \tag{13}$$

where $L_j(k) = \frac{\partial T_j(k)}{\partial W_T^j(k)}$ and $L_j(0) = 0$

Therefore, each element

$$W_T^j(k) = W_F^j(k-1) - \triangle W_F^j(k-1) \tag{15}$$

Where

$$W_F^j(k) = -\eta e_i(k) W_0^j(k)\left[1 - T_j^2(k)\right]\left[T_j(k-1) + W_T^j(k)L_j(k-1)\right] \tag{16}$$

4.5.3 Tuning of input weights

The update equation for input weight vector is also formulated using the above concept. For any j th value, the equation can be given as:

$$W_I^j(k) = W_I^j(k-1) - \triangle W_I^j(k-1) \qquad (17)$$

$$\triangle W_I^j(k) = -\eta e_i(k) W_0^j(k)\left[1 - T_j^2(k)\right]\left[x_{ij}(k) + W_F^j Q_j(k-1)\right] \qquad (18)$$

Where $Q_j(k) = \frac{\partial T_j(k)}{\partial W_I^j(k)}$ and $Q_j(0) = 0$

CHAPTER 5
Simulation Study (Part 1)

Post mathematical discussion, we now evaluate the efficiency of the proposed LRNNIFT based prediction strategy by implementing it on a plant and comparing the results with FFNN and ENN based prediction strategies. To conduct exhaustive analysis, we have considered a complex dynamic plant. In the below example, we have taken only a single hidden layer, 3 hidden neurons, equal learning rate and equal number of epochs in all the identifiers. The reason why we considered similar parameters for the proposed LRNNIFT based identification model and the other two models (ENN, FFNN) is to form a basis for comparison.

Consider a nonlinear time-delayed dynamical system whose dynamics are not known and is represented as mentioned below:

$$y_p(k) = F[x_1, x_2, x_3, x_4, x_5] \tag{19}$$

Where $x_1 = y_p(k-1)$, $x_2 = y_p(k-2)$, $x_3 = r(k-1)$, $x_4 = r(k-2)$, $x_5 = r(k-3)$ where the form of the unknown function is expressed as below:

$$y_p(k) = 0.72x_1 + 0.025x_2x_3 + 0.001{x_4}^2 + 0.2x_5 \tag{20}$$

The identification structure of FFNN, ENN and LRNNIFT is same and is given by following equation

$$Y_{IFT}(k) = \hat{F}[x_1, x_3] \tag{21}$$

In the above case, all the parameters of all three identifiers are trained by each epoch. The number of hidden neurons and the number of inputs for all three identifiers are 3 and 2, respectively. The total number input-output training data was 500 and the training was done for 500 epochs, post which it was graphically observed that the models were successfully trained. The learning rate η

was set to a constant value of 0.0025 for each variable of the identifier. For the training, we have considered the external input r(k) to be a bounded sinusoidal signal with a range of [−1, 1]. It is defined as

$$r(k) = sin\left(\frac{2\pi k}{250}\right)$$

5.1 Discussion on Simulation Results

Fig.6 and Fig.7 show the responses of identifiers during the initial and final stages of training. Evidently, we can conclude that the LRNNIFT has successfully predicted the plant behaviour better than the ENN and FFNN identifiers.

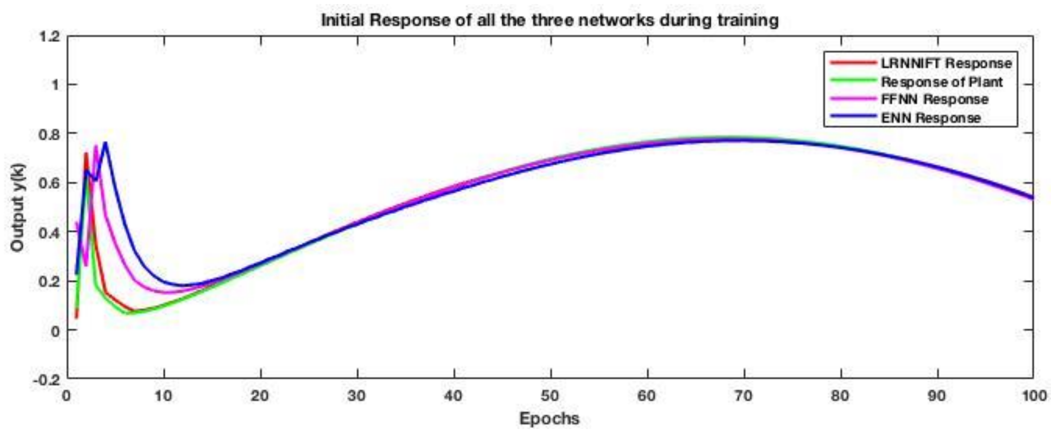


Fig 6: Response of identifiers during initial stages of training

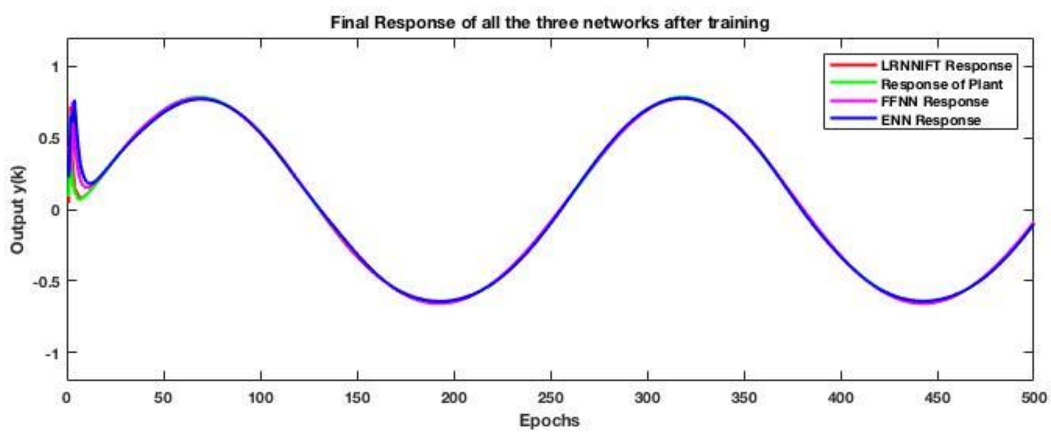


Fig 7: Response of identifiers post training stage

Upon observing the MSE (Mean Square Error) and MAE (Mean Average Error) plots (shown in Figs.8-9) obtained during the training, we can again deduce that the LRNNIFT identifier has given the lowest value of error along-with fastest convergence of instantaneous error among all three identifiers.

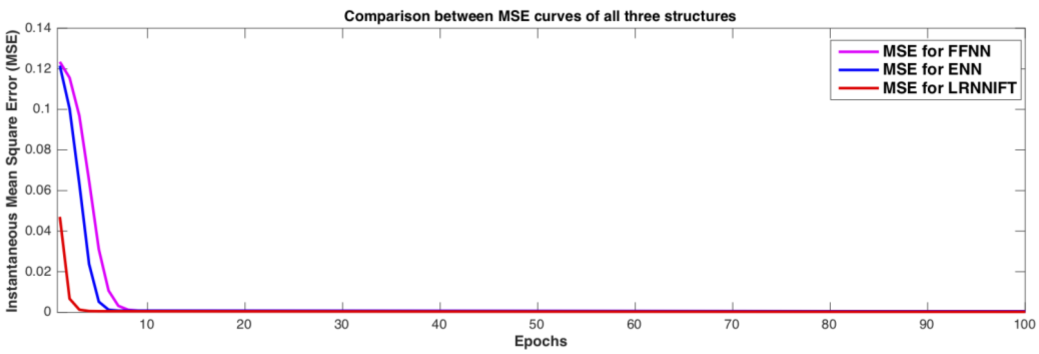


Fig 8: Comparison of MSE curves of all three identifiers

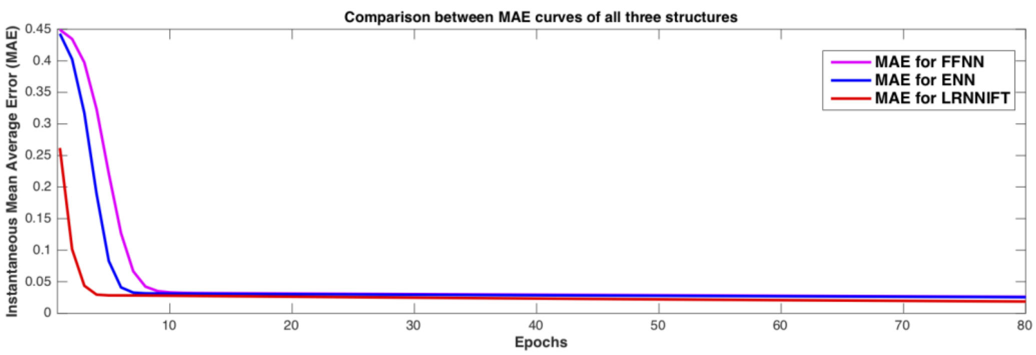


Fig 9: Comparison of MAE curves of all three identifiers

The numerical observations obtained from the MSE and MAE plots are given in form of comparison in Table 1. From the table, the average MSE (AMSE) and total MAE (TMAE) values are found to be the lowest in case of LRNNIFT identifier. We can also observe that with the same number of inputs, LRNNIFT model is performing better both in terms of error reduction and rate of convergence. This makes it more computationally efficient than FFNN and ENN.

Table 1: Comparison of performance of LRNNIFT, ENN & FFNN identifiers

Neural network (No. of epochs=500)	ENN	FFNN	LRNNIFT
Average MSE	8.65×10^{-5}	5.19×10^{-4}	1.5659×10^{-5}
Total MAE	0.011	0.0083	0.0047

5.2 Validation Stage

Once the training of models is completed, the models are set to be tested with different inputs to the plant. This stage is important because it puts the quality of training on check and we can observe the actual efficiency of the identifier. In other words, this stage categorizes how well the plant is trained by the sample so that it can successfully predict output for data outside its sample space. Here, we take a different external input (having the same range as that of the input used during training). The input subjected to the plant is varied in nature to check the versatility of the plant's prediction capabilities.

The external input is given as follows:

If k is greater than 4 and less than or equal to 250,

$$r(k) = sin\left(\frac{\pi k}{25}\right)$$

If k is greater than 250 and less than or equal to 500,

$$r(k) = 1$$

If k is greater than 500 and less than or equal to 750,

$$r(k) = -1$$

If k is greater than 750,

$$r(k) = 0.3sin\left(\frac{\pi k}{25}\right) + 0.1sin\left(\frac{\pi k}{32}\right) + 0.6sin\left(\frac{\pi k}{10}\right)$$

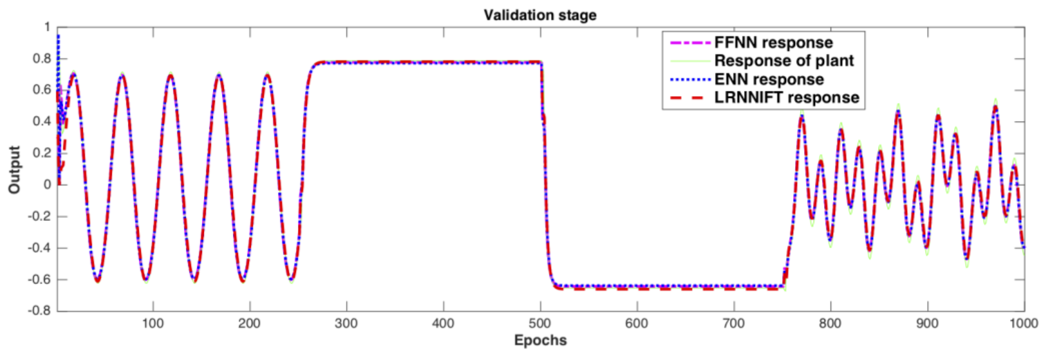


Fig 10: Comparison of plant output with identifier responses during validation

The validation response plots of all three identifiers can be seen in Fig.10. It can be observed that all three identifiers have been trained successfully with LRNNIFT showing superior results.

CHAPTER 6

Adaptive Control using LRNNIFT

After successfully testing the LRNNIFT structure as an identifier, we move on to the second part of the analysis. Based on the proposed structure, a controller is designed to control such nonlinear dynamical systems. Output of the controller will act as the corrected input to the plant.

6.1 Controller Structure

The structure of the controller (similar to the identifier) is shown below.

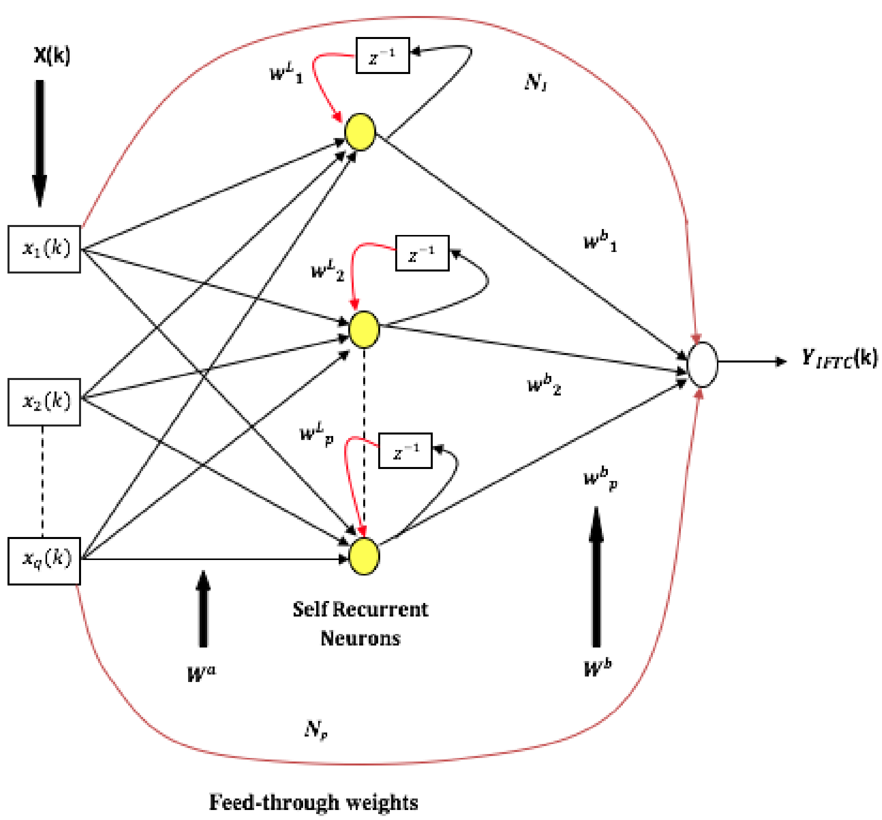


Fig 11: Structure of the proposed controller

In figure 11, the bright red arrows represent the local recurrent weights generated as output of hidden neuron or node and propagated through a lag of a unit instant as connected back to the same neuron. This leads to the formation of a locally recurrent structure. The maroon arrows represent feed through weights connecting the input layer neurons to the output layer node through weights denoted as N= {N_1, N_2...N_q}. The recurrent weights are defined as W_L = {w_1, w_2...w_p} while W_a represents the input weight vector. All weights can be updated. From the figure, it can be seen that if W_L and N are removed or made equal to, the structure is reduced to a FFNN. Furthermore, FFNN and ENN structures are considered to compare the LRNNIFT controller structure. The reason behind these structures to be chosen for comparison was primarily for their proven performance in the fields of both estimation and control.

FFNN is a simple structure with no recurrent weights whereas ENN has a rather complex structure with every recurrent weight fed back to each hidden node.

The output weight vector is given as W_b = {w_b^1, w_b^2....w_b^p}.

The input vector is defined as X= {x_1, x_2....x_q}. Subsequently, the output of any p^{th} recurrent node at any k^{th} instant can be calculated as:

$$O_p(k) = [f U_p(k)] \tag{22}$$

The function f is a hyperbolic tangent function.

The induced field (IF) of any p^{th} recurrent node can be given as:

$$U_p(k) = W_p^L(k) O_p(k-1) + \sum_q W_{pq}^b(k) x_q(k) \tag{23}$$

The IF of the output node is described below:

$$S(k) = \sum_p W_b^p(k) O_p(k) \tag{24}$$

The output value from the output neuron will be equal to the sum of its own IF and the feed-through factor (from input) as a linear function has been considered as the activation function. Hence, we write it as:

$$S_{IFTC}(k) = S(k) = \sum_p W_0^p(k) U_p(k) + \sum_q N_{pq}(k) x_q(k) \tag{25}$$

6.2 Indirect Adaptive Control of a time-delayed nonlinear system

As discussed previously, designing a controller for a system which is dynamic and nonlinear in nature is a complicated task as linear control techniques fail on such plants. Artificial neural networks have majorly solved this problem due to their flexible nature as there is a vast majority of structures from which an appropriate structure can be chosen whose parameters can be tuned based on system requirements. In this brief, a modified recurrent neural network is used as a controller for a dynamic plant which is to be controlled along a reference model. The general mathematical formulation of a nonlinear time delayed plant can be given as:

$$Y_{LRN}(k) = F[Y_{LRN}^{q}(k-1), Y_{LRN}^{q}(k-2) \ldots Y_{LRN}^{q}(k-O), u_c(k-1), u_c(k-2), u_c(k-D)]$$

In the above equation, $Y_{LRN}^{q}(k-1)$ represents a previous value of the system delayed by an instant. In a similar fashion, all the outputs of $Y_{LRN}(k)$ are mentioned till q^{th} instant. Similarly, input past values are written $u_c(k-1)$ to $u_c(k-D)$. Here, $u_c(k)$ is essentially the controller output which will act as an input to the plant. The actual input which will be fed to the controller is $r(k)$.The aim here is to control the above plant, that is, to align its response with the reference model (desired response of the plant). This simply translates to $F \approx F_m$, where $F_m$is the reference model. The potency of any nonlinear control method is established only if it reduces the dependency on plant parameters and structural complexity along-with providing faster control response. Therefore, in case of LRNNIFT controller, only three inputs are taken from the vast array of system variables- the present value of input to plant, $u_c(k)$, one previous value of output, $Y_{LRN}^{q}(k-1)$, and a previous value of external input, $r(k-1)$. The control scheme is estimated relying on these three inputs only and is calculated as $Y_{LRN}(k)$. Motivation behind selecting few inputs (here, three) is that this minimizes plant parameter dependency of the controller along with reducing the computation load by lowering the number of weights to be adjusted. Figure 12 represents the block diagram of the proposed LRNNIFT controller.

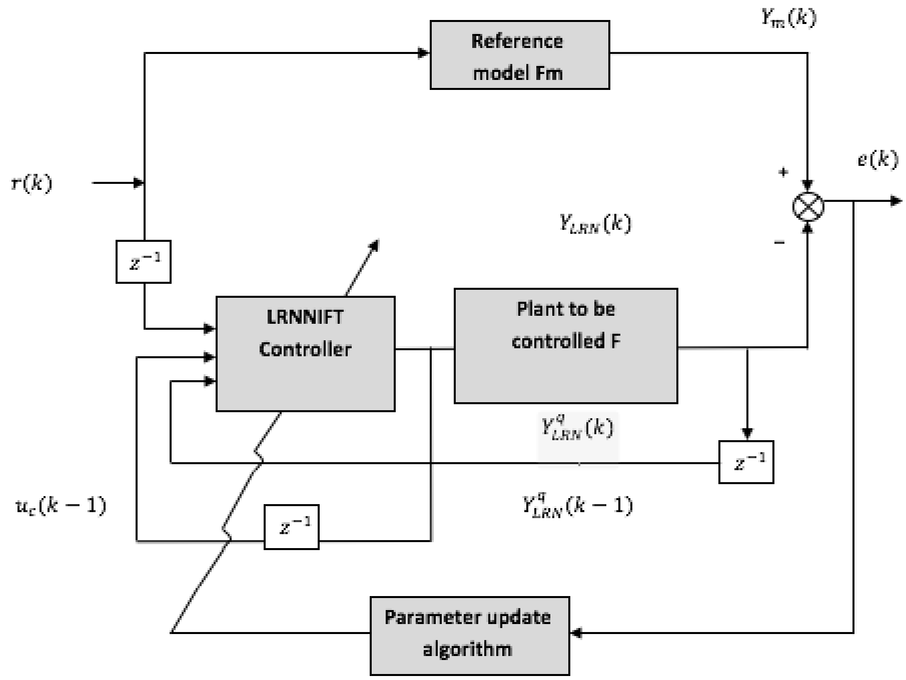


Fig 12: Adaptive control scheme using LRNNIFT model (Proposed)

6.3 Learning Algorithm for LRNNIFT controller

The versatile back propagation method is applied for error minimization during incremental training. Let *e(k)* denote the instantaneous error between the output of plant and reference model at kth instant and is given as:

$$E(k) = \frac{1}{2}[Y_m(k) - Yc(k)]^2 = \frac{1}{2}e^2(k) \tag{26}$$

6.3 Tuning of Weights

6.3.1 Tuning of output and feed-through weights

The error gradient with respect to any j^{th} output layer weight can be expressed as

$$\frac{\partial E(k)}{\partial W_b^j(k)} = \left(\frac{\partial E(k)}{\partial Y_c(k) \times \frac{\partial Y_c(k)}{\partial W_b^j(k)}} \right) \tag{27}$$

where $\frac{\partial Y_c(k)}{\partial W_b^j(k)} = O_j(k)$

Thus, each value in $W_b(k) = w_b^1(k), w_b^2(k), \ldots . w_b^q(k)$ will be updated to a new value as following:

$$W_b(k) = W_b^j(k-1) - \triangle W_b^j(k-1) \tag{28}$$

Where $\triangle W_b^j(k-1) = -\eta J(k)e(k-1)O_j(k-1)$

In a similar manner for the feed through weights, each value in N= {N_1, N_2...N_q} will be updated to a new value as following:

$$N^j(k) = N^j(k-1) - \triangle N^j(k-1) \tag{29}$$

where $\triangle N_0^j(k-1) = -\eta J(k)e(k-1)O_j(k-1)$

where J(k) is the Jacobian for the plant.

4.5.2 Tuning of locally recurrent weights

The update equation for any j th recurrent weight vector can be written as below:

$$\frac{\partial E(k)}{\partial W_L^j(k)} = \left(\frac{\partial E(k)}{\partial Y_{'(k)} \times \frac{\partial Y_c(k)}{\partial u_c(k)} \times \frac{\partial u_c(k)}{\partial S(k)} \times \frac{\partial S(k)}{\partial W_L^j(k)}} \right) \tag{30}$$

where $\frac{\partial Y_c(k)}{\partial W_L^j(k)} = \frac{\partial Y_c(k)}{\partial S(k)} \frac{\partial S(k)}{\partial W_L^j(k)}$

Now, $\frac{\partial Y_c(k)}{\partial S(k)} = 1$

Hence, evaluation of $\frac{\partial S(k)}{\partial W_L^j(k)}$ is as below:

$$\frac{\partial S(k)}{\partial W_L^j(k)} = \frac{\partial V(k)}{\partial O_j(k)} \frac{\partial O_j(k)}{\partial W_L^j(k)} \tag{31}$$

Where $\frac{\partial V(k)}{\partial O_j(k)} = W_b^j(k)$ and,

$$\frac{\partial O_j(k)}{\partial W_L^j(k)} = \left(1 - O_j^2(k)\right)[O_j(k-1) + W_L^j(k) L_j(k-1)] \tag{32}$$

where $L_j(k) = \frac{\partial O_j(k)}{\partial W_o^j(k)}$ and $L_j(0) = 0$

Therefore, each element

$$W_L^j(k) = W_L^j(k-1) - \triangle W_L^j(k-1) \tag{33}$$

Where

$$W_L^j(k) = -\eta J(k) e(k) W_b^j(k) \left[1 - O_j^2(k)\right]\left[O_j(k-1) + W_L^j(k) L_j(k-1)\right] \tag{34}$$

4.5.3 Tuning of input weights

The update equation for input weight vector is also formulated using the above concept. For any j^{th} value, the equation can be given as:

$$W_a^j(k) = W_a^j(k-1) - \triangle W_a^j(k-1) \tag{35}$$

$$\triangle W_a^j(k) = -\eta J(k) e(k) W_b^j(k) \left[1 - O_j^2(k)\right]\left[x_{ij}(k) + W_L^j Q_j(k-1)\right] \tag{36}$$

Where $Q_j(k) = \frac{\partial O_j(k)}{\partial W_a^j(k)}$ and $Q_j(0) = 0$

CHAPTER 7

Simulation Study (Part 2)

In order to evaluate the efficacy of the proposed LRNNIFT based control strategy, the scheme is implemented on a complex dynamic system. Furthermore, the results obtained from the proposed controller are compared with the FFNN and ENN controllers. Structurally, a single input, hidden and output layer, 4 hidden neurons, uniform learning rate and instantaneous training is applicable to all the three controllers. The reason why we considered uniformity among structure parameters for our analysis is to better judge the performance of the LRNNIFT controller.
For simulation, the following nonlinear dynamical plant has been considered:

$$y_0(k) = \frac{y_0(k-1)y_0(k-2)[y_0(k-1)+2.5]}{1+y_0^2(k-1)+y_0^2(k-2)} + u_c(k-1) \tag{37}$$

Where $u_c(k)$ denotes the input to the plant. The reference model is given as:

$$y_m(k) = 0.6y_m(k-1) + 0.3y_m(k-2) + r(k) \tag{38}$$

Where $r(k)$ is the BIBO stable external input to the system given as:

$$r(k) = sin\left(\frac{\pi k}{25}\right) \tag{39}$$

The control objective here is to bring the difference between reference model and plant's response $e_c(k) = y_m(k) - y_0(k)$ approximately equal to zero by introducing an optimal control signal $u_c(k)$ at every instant, to the plant via LRNNIFT as a rectified input to it. $u_c(k)$ can be computed from the knowledge of $y_0(k)$ and its past values as

$$u_c(k) = F[(y_o(k), y_o(k-1)] + 0.6y_m(k-1) + 0.3y_m(k-2) + r(k)$$

For our analysis, the following parameter values have been considered:

Learning rate = 0.028

Total number of hidden neurons = 4

7.1 Discussion on Simulation Results

Figure 13 represents the plant output response (in dotted pink) along with reference model response (in solid green) without control scheme implementation. From the plot, it can be clearly observed that the two responses do not coincide (as desired). Therefore, we use the adaptive control configuration shown in Fig. 12 and apply it to the plant.

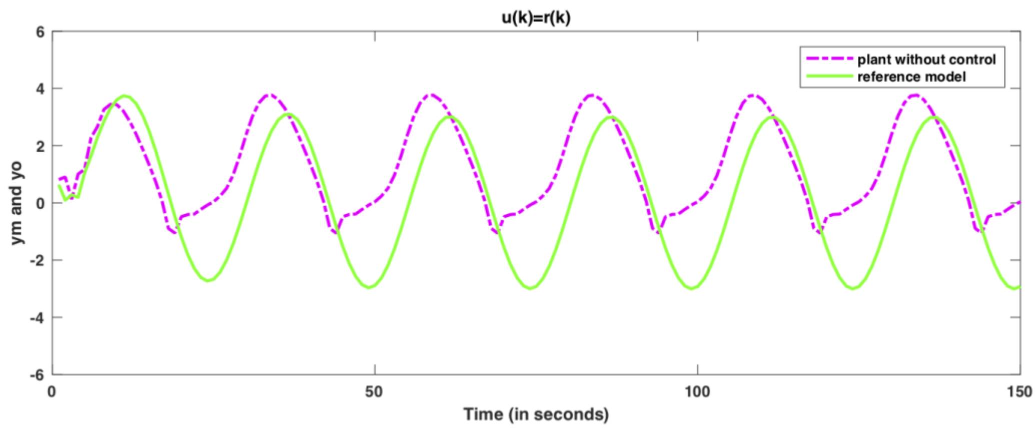


Fig 13: Plant response without controller

Figure 14 shows the response of LRNNIFT controller compared with FFNN and ENN based controllers and plant during the early stages of training. The instantaneous training was done for 60,000-time steps after which it was terminated. Post training, the controllers started tracking the reference plant's output which can be seen in figure 15. From figure 14, we can clearly observe that the LRNNIFT controller has the fastest response among all three. Additionally, it is able to force the plant to track the reference model from the very first instant. Time of response being a critical aspect in controller design makes the proposed controller better than ENN and FFNN based control. Table 2 shows the Average Mean Square Error (AMSE) and Total Mean Average Error values for all the three controllers which is also the least for the proposed controller.

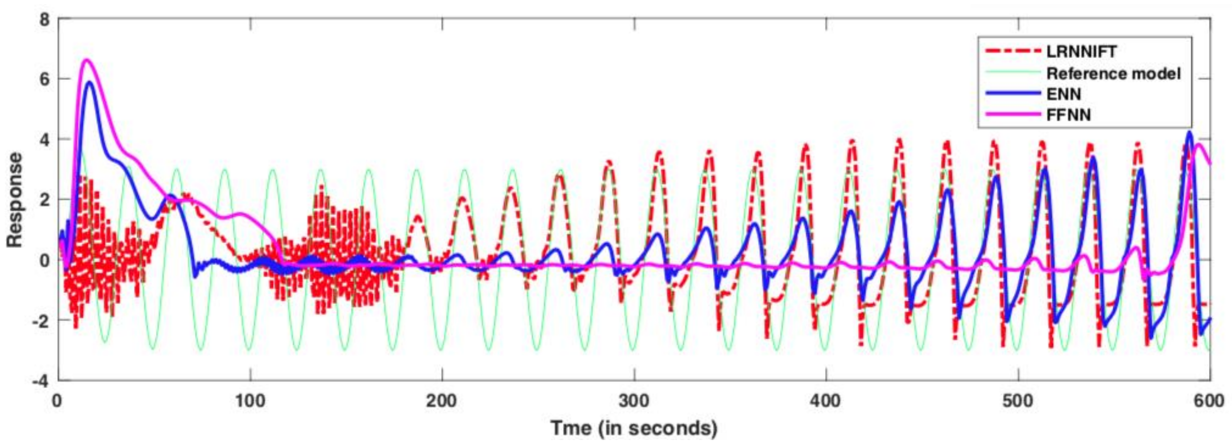


Fig 14: Response of controllers during initial phases of training

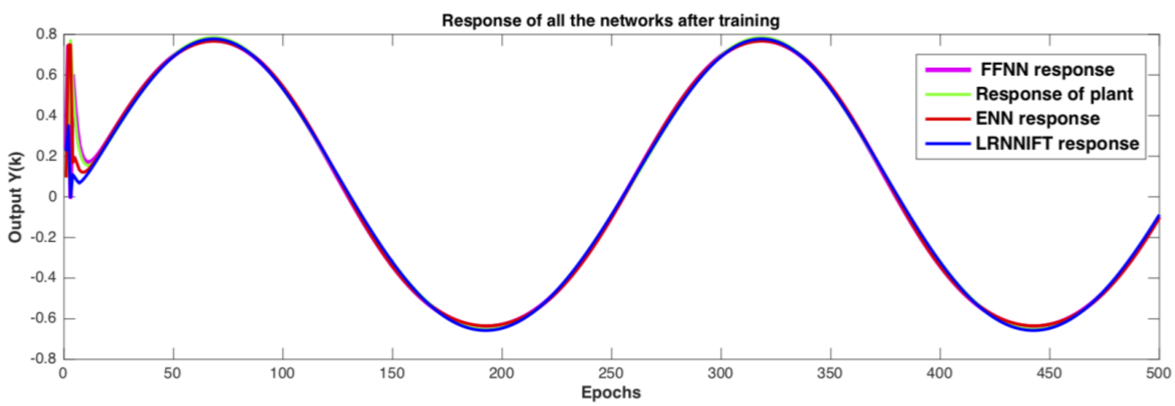


Fig 15: Response of controllers after successful training

Table 2: Comparison of performance of LRNNIFT, ENN & FFNN controllers

Parameters	**Performance Comparison Table**		
	ENN	**FFNN**	**LRNNIFT**
Average MSE	0.0469	0.0907	0.0453
Total MAE	0.1329	0.1956	0.1261

7.2 Disturbance rejection test

The proposed controller is also checked for robustness against disturbance signals in the system. This is one of the key aspects of closed loop control.

A step signal of amplitude 5 is added as disturbance to the plant at k=55000th instant.

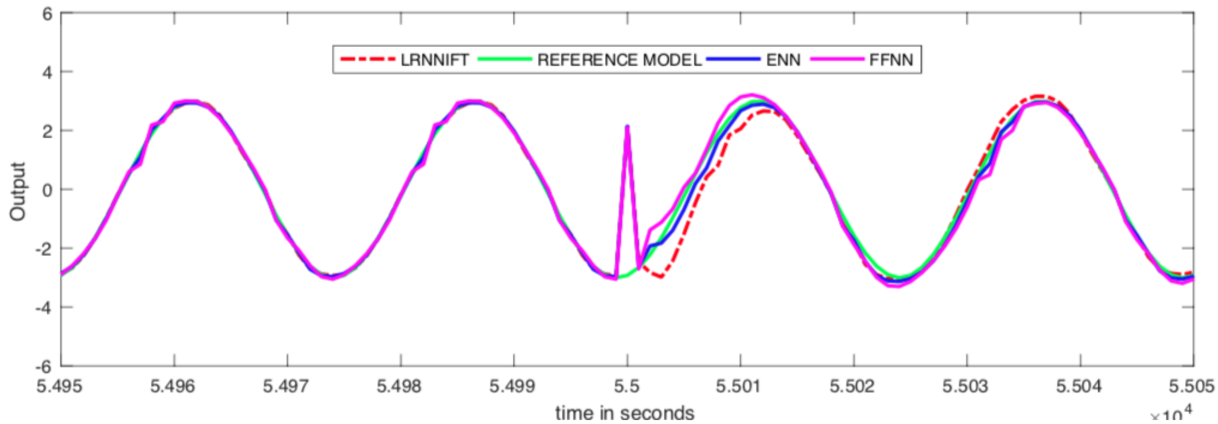


Fig 16: Disturbance Analysis for robustness

The disturbance leads to a spike in controller response and the instantaneous mean square errors and mean average errors also experience the same. In figure 16, we can see the noise signal causing disturbance at k=55000 instant but as the training went on it rapidly recovered and went back on the track within few instants. This proves the robust or adaptive nature of the controller. On comparison, we can observe that the FFNN controller has under-performed whereas the ENN controller has performed only slightly better than the LRNNIFT controller. This is because the ENN is an extremely complex structure, that is- for equal number of inputs and hidden neurons, the number of update parameters (or weights) for ENN is 32 whereas for LRNNIFT it is only 20. FFNN controller has 16 weights only but it is slow and less accurate as it cannot track the plant's past values.

CONCLUSION

In this thesis, an alternative neural network is proposed for prediction of nonlinear systems with unknown dynamics. The LRNNIFT structure developed consists of locally recurrent loops along with input fed through weights directly to the output. The error back-propagation method is implemented for tuning of parameters and error reduction by optimizing the cost function. The model is tested on a complex nonlinear system and compared with feed forward (FFNN) and Elman neural network (ENN) identification models. From the simulation study, it is observed that this model performs better in terms of learning dynamics for fewer inputs when compared to the other two reference models. Although the results of ENN and FFNN are comparable, despite the slightly high error in ENN model, we can observe that it has a better convergence rate as compared to FFNN. In further continuation to above, an adaptive controller is designed based on this network for nonlinear plants. Parameter tuning is again done via back-propagation method. The controller is implemented on a nonlinear complex system and its results are compared with FFNN and ENN controllers. The simulation results clearly depict that the proposed controller performs better than the other two controllers both in terms of error mitigation and speed of tracking. The controllers are also tested for robustness by introducing a disturbance signal in the plant equation. It is observed that the proposed controller successfully adapts by moving back to the original track. Although the results of ENN are slightly better than LRNNIFT in terms of robustness, the drawback here would be the high complexity of the ENN network which again leads to the proposed controller to be a better choice. After extensive mathematical analysis and simulation results, we can conclude that the proposed network, LRNNIFT, can be regarded as a general identification model that can be applied to a wide class of nonlinear dynamic systems for the purpose of behavior prediction as well as provides better control over such plants.

REFERENCES

[1] K. Li, H. Luo, S. Yin and O. Kaynak, "A Novel Bias-Eliminated Subspace Identification Approach for Closed-Loop Systems," in *IEEE Transactions on Industrial Electronics*, vol. 68, no. 6, pp. 5197-5205, June 2021, doi: 10.1109/TIE.2020.2989717.

[2] H. Fan, M. Wu, X. Lai, S. Du, C. Lu and L. Chen, "Process monitoring based on probabilistic principal component analysis for drilling process," *IECON 2021 – 47th Annual Conference of the IEEE Industrial Electronics Society*, 2021, pp. 1-6, doi: 10.1109/IECON48115.2021.9589495.

[3] L. Li, F. Wang, H. Zhang and X. Ren, "Adaptive Model Recovery Scheme for Multivariable System Using Error Correction Learning," in *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1-10, 2021, Art no. 3001710, doi: 10.1109/TIM.2021.3108569.

[4] S. Yin, X. Li, H. Gao and O. Kaynak, "Data-Based Techniques Focused on Modern Industry: An Overview," in *IEEE Transactions on Industrial Electronics*, vol. 62, no. 1, pp. 657-667, Jan. 2015, doi: 10.1109/TIE.2014.2308133.

[5] Sanger, Terence D. "Optimal unsupervised learning in a single-layer linear feedforward neural network." Neural networks 2.6 (1989): 459-473.

[6] Svozil, Daniel, Vladimir Kvasnicka, and Jiri Pospichal. "Introduction to multi-layer feed-forward neural networks." Chemometrics and intelligent laboratory systems 39.1 (1997): 43-62

[7] -T. Lee, J.-T. Jeng, The Chebyshev-polynomials-based unified model neural networks for function approximation, IEEE Trans. Syst. Man Cybern. Part B (Cybern.) 28 (6) (1998) 925–935

[8] Medsker, Larry R., and L. C. Jain. "Recurrent neural networks." Design and Applications 5 (2001): 64-67.

[9] Q.H. Cao, S.A. Wang, Identification of nonlinear dynamic system based on El- man neural network, Comput. Eng. Appl. 43 (31) (2006) 87–89.

[10] X. Zhang, R. Wang, T. Liao, T. Zhang, Shortterm forecasting of wind power generation based on the similar day and Elman neural network, in: Computational Intelligence, IEEE Symposium, 2015, pp. 647–650.

[11] P. Li, et al., Application of a hybrid quantized Elman neural network in short- -term load forecasting, Int. J. Electr. Power Energy Syst. 55 (2) (2014) 749–759. [18] J. Kennedy, R. Eberhart, Particle swarm optimization, in: IEEE International

[12] Conference on Neural Networks, 4, 1995, pp. 1942–1948. Y.X. Su, R. Chi, Multi-objective particle swarm-differential evolution algorithm,

[13] P. Ghamisi, J.A. Benediktsson, Feature selection based on hybridization of genetic algorithm and particle swarm optimization, IEEE Geosci. Remote Sens. Lett. 12 (2) (2014) 309–313.

[14] Nelles, Oliver. Nonlinear System Identi cation: From Classical Ap- proaches to Neural Networks, Fuzzy Models, and Gaussian Processes. Springer Nature, 2020.

[15] Neural Comput. Appl. 28 (2) (2017) 407–418.

[16] Ha ̈gglund, Tore, and K. J. A ̊stro ̈m. ”REVISITING THE ZIEGLER- NICHOLS TUNING RULES FOR PI CONTROL—PART II THE FREQUENCY RESPONSE METHOD.” Asian Journal of Control 6.4 (2004): 469-482.

[17] J. J. Gude and E. Kahoraho, "Modified Ziegler-Nichols method for fractional PI controllers," 2010 IEEE 15th Conference on Emerging Technologies Factory Au- tomation (ETFA 2010), 2010, pp. 1-5, doi: 10.1109/ETFA.2010.5641074.

[18] Wang, Qing-Guo, Tong-Heng Lee, and Yu Zhang. "Multiloop Version of the Modified Ziegler Nichols Method for Two Input Two Output Processes." Industrial engineering chemistry research 37.12 (1998): 4725-4733.

[19] Lorenzini, Charles, Luís Fernando Alves Pereira, and Alexandre Sanfelice Bazanella. "A generalized forced oscillation method for tuning proportional-resonant con- trollers." IEEE Transactions on Control Systems Technology 28.3 (2019): 1108-1115.

[20] Bazanella, Alexandre Sanfelice, Luís Fernando Alves Pereira, and Adriane Parraga. "A new method for PID tuning including plants without ultimate frequency." IEEE Transactions on Control Systems Technology 25.2 (2016): 637-644.

[21] Cetin, Meric, and Serdar Iplikci. "A novel auto-tuning PID control mechanism for nonlinear systems." *ISA transactions* 58 (2015): 292-308.

[22] Hunt, Kenneth J., et al. "Neural networks for control systems—a survey." Automatica 28.6 (1992): 1083-1112.

[23] Isidori, Alberto, E. D. Sontag, and M. Thoma. Nonlinear control systems. Vol. 3. London: springer, 1995.

[24]Hunt,KennethJ.,etal.”Neural Networks For Control Systems—a survey.” Automatica 28.6 (1992): 1083-1112.

[25] Bloch, Ge´rard, and Thierry Denoeux. ”Neural networks for process control and optimization: Two industrial applications.” ISA transactions 42.1 (2003): 39-51.

[26] Savran, Aydogan, Ramazan Tasaltin, and Yasar Becerikli. ”Intelligent adaptive nonlinear flight control for a high performance aircraft with neural networks.” ISA transactions 45.2 (2006): 225-247

[27] Perrusquía, Adolfo, and Wen Yu. "Identification and optimal control of nonlinear systems using recurrent neural networks and reinforcement learning: An overview." Neurocomputing 438 (2021): 145-154.

[28] X. Tai, L. Wang, Prediction of Short-Term Power Load Based on Elman Neural Network and Fuzzy Control, World Sci-Tech R & D, 2016.

[29] T. Wang, H. Gao and J. Qiu, ”A Combined Adaptive Neural Network and Nonlinear Model Predictive Control for Multirate Networked In- dustrial Process Control,” in IEEE Transactions on Neural Networks and Learning Systems, vol. 27, no. 2, pp. 416-425, Feb. 2016, doi: 10.1109/TNNLS.2015.2411671.

[30] Isidori, Alberto, E. D. Sontag, and M. Thoma. Nonlinear control systems. Vol. 3. London: springer, 1995.

[31] X.Z. Gao, X.M. Gao, S.J. Ovaska, A modified Elman neural network model with application to dynamical systems identification, IEEE Int. Conf. Syst. Man Cy- bern. 2 (2) (1996) 1376–1381.

[32] R. Chi, et al., A hybridization of cuckoo search and particle swarm optimization for solving optimization problems, Neural Comput. Appl. doi:10.1007/ s00521-017-3012-x.

[33] Ding, Feng, and Tongwen Chen. ”Identi cation of Hammerstein non- linear ARMAX systems.” Automatica 41.9 (2005): 1479-1489.

[34] Kubat, Miroslav. "Neural networks: a comprehensive foundation by Simon Haykin, Macmillan, 1994, ISBN 0-02-352781-7." *The Knowledge Engineering Review* 13.4 (1999): 409-412.

[35] Sazli, Murat H. "A brief review of feed-forward neural networks." *Communications Faculty of Sciences University of Ankara Series A2-A3 Physical Sciences and Engineering* 50.01 (2006).

[36] Kumar, Rajesh, et al. "Diagonal recurrent neural network based identification of nonlinear dynamical systems with Lyapunov stability based adaptive learning rates." *Neurocomputing* 287 (2018): 102-117.

[37] Johansson, Erik M., Farid U. Dowla, and Dennis M. Goodman. "Backpropagation learning for multilayer feed-forward neural networks using the conjugate gradient method." *International Journal of Neural Systems* 2.04 (1991): 291-301.

[38] Rizk-Allah, Rizk M., and Aboul Ella Hassanien. "COVID-19 Forecasting Based on an Improved Interior Search Algorithm and Multilayer Feed-Forward Neural Network." *Medical Informatics and Bioimaging Using Artificial Intelligence*. Springer, Cham, 2022. 129-152.

[39] Singh, Krishna Veer, Hari Om Bansal, and Dheerendra Singh. "Fuzzy logic and Elman neural network tuned energy management strategies for a power-split HEVs." *Energy* 225 (2021): 120152.

[40] Li, Xiuming, et al. "Modeling for indoor temperature prediction based on time-delay and Elman neural network in air conditioning system." *Journal of Building Engineering* 33 (2021): 101854.

LIST OF PUBLICATIONS

Conference Name	Paper Title	Paper Status	Scopus Indexed
Delcon 2022: 1st EEE Delhi Section International Conference on Electrical, Electronics and Computer Engineering	A Novel Recurrent Neural Network Model for the Identification of Time-Delayed Nonlinear System	Published in 2022 IEEE Delhi Section Conference (DELCON) Link to paper: https://ieeexplore.ieee.org/document/9753622 Date of Conference: 11th-13th February, 2022	Yes
Sigma 2022: 2nd International Conference on Signals, Machine, and Automation	Indirect Adaptive Control of Nonlinear system using Recurrent Neural Network	Accepted Date of Conference: 5th and 6th August, 2022 Will be published as **proceedings with Springer** in their prestigious "**Lecture Notes in Electrical Engineering**" series (https://www.springer.com/series/7818)	Yes

A Novel Recurrent Neural Network Model for the Identification of Time-Delayed Nonlinear System

Prashasti Srivastava
Department of Electrical Engineering
Delhi Technological University
Delhi, India
prashastisrivastava@dtu.ac.in, prashastips@gmail.com

Rajesh Kumar
Department of Electrical Engineering
Delhi Technological University
Delhi, India
rajeshkumar@dtu.ac.in, rajeshmahindru23@gmail.com

2022 IEEE Delhi Section Conference (DELCON) | 978-1-6654-5883-2/22/$31.00 ©2022 IEEE | DOI: 10.1109/DELCON54057.2022.9753622

***Abstract*—This paper presents a locally recurrent neural network with input feed through (LRNNIFT) for the identification of nonlinear time-delayed system. The proposed network model parameters are tuned using a Back-propagation (BP) algorithm. The performance of the proposed model is compared with the well known recurrent Elman neural network (ENN) and a single layer feed-forward neural network (FFNN). The simulation results showed that the proposed model has shown better accuracy as compared to the other two models.**

***Index Terms*—locally recurrent neural network with input feed through, Recurrent neural network, Identification, Elman neural network, back-propagation method**

I. Introduction

In the ever-growing world of science, extensive research is being performed in understanding the behavior of the universe and its systems. This has led to the development of various types of mathematical formulations. The significance of nonlinear system identification is unarguable as most control methods are designed to perform in the linear region only and have a restricted range of working [1]. It is evident that linear control methods are popular because they are simple but their effectiveness in real-world (nonlinear) system control is very bounded. Conventional control methods such as the Zieglar Nichols method is found to be inadequate when it is mandatory to control systems in real time. The linear identification process involves the selection of a specific appropriate phase of the model in order to best measure the behavior of a system[2–4]. To overcome this problem, researchers started focusing on modeling techniques for control of such systems. Nonlinear system modeling techniques are versatile in nature and can work in a vast range unlike linear identification methods. Artificial neural networks have been repeatedly implemented in the field of nonlinear control systems for their inherent nonlinear behaviour. This has given rise to studies on various models for dynamical systems control and identification. It was observed that static (feed forward) neural networks are not efficient enough for control engineering applications, since they do not have the dynamical attribute. Consequently, recurrent neural networks have been regarded more importance in the recent years for the same [5-7].

978-1-6654-5883-2/22/$31.00 ©2022 IEEE

In compliance with the above approach, in this paper we discuss a locally recurrent neural network which has a set of weights directly fed to the output node from the external input. Our purpose here is not to exhaustively search for the best possible results, but to compare output of this novel structure with existing feed forward neural network and elman neural network models maintaining uniformity among all parameters. This is done to observe the true behavior of the structure taking the preexisting models as a reference for nonlinear system identification.

II. Artificial Neural Networks-Brief overview

An artificial neural network or ANN is an algorithm loosely based on the working of human brain. Just as the brain consists of thousands of neurons and synapses associated with them, an artificial neural network also contains a set of nodes (known as neurons) that are interconnected via weights. In a sense, the weights determine the degree to which the information is to be transmitted to another node. Hence, they play a vital role in the working of a neural network. These weights are updated during the training process using an error reduction algorithm to produce desired output. Neural networks are used as black box models as they require only the input and output data set to draw patterns and to estimate the mathematical relationship between the input and output of a system. On a broad structural classification, artificial neural networks can be of two sorts: the feed forward neural network (FFNN) and the recurrent (or repetitive) neural network (RNN) [8]. Both of these networks contain characteristics which makes them an ideal candidate to use flexible real-time dynamic controls [9–11].

III. Structure of LRNNIFT

LRNNIFT or locally recurrent neural network with input feed through is represented in Fig.1. In the figure, the green arrows in the structure of LRNNIFT, depict the local weights which propagate a unit delay as output of the hidden neuron which is looped back as an input to the same hidden neuron. This forms the locally recurrent structure. The pink arrows denote the input fed to the output through feed through weights represented as M= M_1, M_2,...M_q. The local weights are given by local weight vector, W_F = w_1, w_2...w_q and W_I

represents the input weight vector having tunable weights. It can be deduced from the figure that if vector M_q is equal to zero, the structure of LRNNIFT reduces to an Elman Neural Network (ENN) and similarly if W^F is made zero, it reduces the structure to a Feed Forward Neural Network (FFNN). Both these structures are used further in this paper to compare the LRNNIFT structure. The output layer and hidden layer neurons are connected by a weight vector given as output weight vector, W_0 = w_0^1, w_0^2....w_0^p . The q-input vector depicts the external signal which is applied to the network and is represented as a vector by X= $x_1(k)$, $x_2(k)$....$x_q(k)$. Additionally, the output of any p^{th} hidden neuron at any k^{th} point in time is given by:

$$T_p(k) = f[Z_p(k)] \tag{1}$$

It is to be noted that f here is taken as the tangent hyperbolic function.

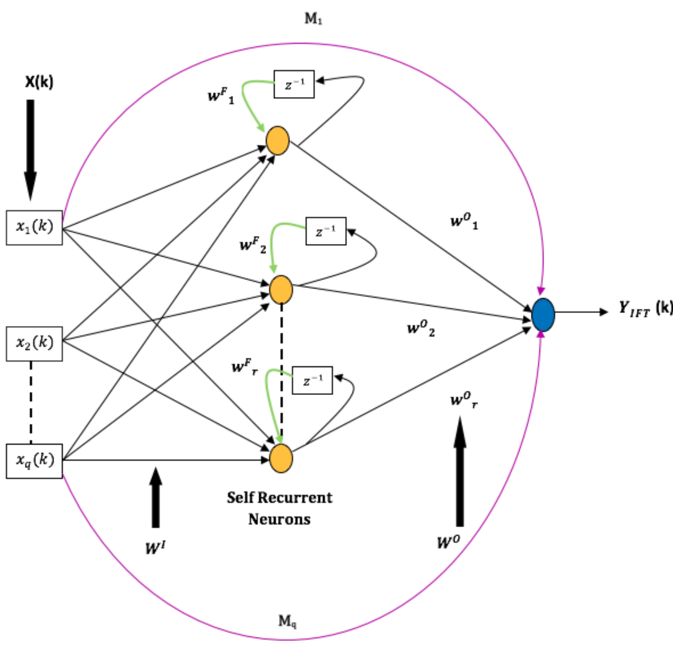


Fig. 1. Structure of locally recurrent neural network with input feed through

The induced field of any p^{th} hidden neuron is calculated as below:

$$Z_p(k) = W_p^F(k)T_p(k-1) + \sum_q W_{pq}^I(k)x_q(k) \tag{2}$$

Further, the induced field of the output neuron is as follows:

$$V(k) = \sum_p W_0^p(k)T_p(k) \tag{3}$$

A linear activation function is considered for the output layer neuron, therefore the output will be equal to its own induced field along with summation of external input via feed through weights. Subsequently, we can deduce:

$$V_{IFT}(k) = V(k) = \sum_p W_0^p(k)T_p(k) + \sum_q M_{pq}(k)x_q(k) \tag{4}$$

A. Identification of time-delayed nonlinear system

In any control scheme, the basic feature is to provide the desired control of plant parameters. It is noteworthy to mention that in order to implement a control strategy on a plant, it is essential to know its dynamics. It is cumbersome to design a controller for plant which does not have a mathematical model. Subsequent to this, a major class of real time systems are present whose dynamics can not be mathematically modelled and are non-deterministic in nature. This is where ANN has played a major role in the recent times by helping estimate the unknown behaviour of nonlinear plants that are to be controlled. The neural networks are backed by several robust learning algorithms which can be modified as per the nature of system in consideration. With respect to this paper, neural networks are used as an identifier for a dynamical system whose properties are partially known. The general mathematical expression of a dynamical system is as follows:

$$\begin{aligned} y_q(k) = F[&y_q(k-1), y_q(k-2)...y_q(k-O), \\ &r(k-1), r(k-2)...r(k-L)] \end{aligned} \tag{5}$$

As mentioned above, equation (5) represents a general dynamical system where Y_q(k-1) is the past value of plant (with a difference of one instant) and similarly all past values of y_q output are represented till q^{th} instant. In the same manner, the past values of input are also represented as r(k-1) to r(k-L). The term O represents the order of the plant and L is generally taken to be less than O. It is to be noted that for the sake of simplicity in understanding, the above equation is represented for k^{th} time instant, that is, if the above equation is shifted by a unit delay, it will be visibly evident that the next value of the plant will depend upon the past as well as the present values of input and output. Therefore, in all further discussions this notion has been kept in mind. Further, the nonlinear system to be estimated is depicted as the function F.

Our prime objective here is to estimate the behaviour of the above plant, that is, our end result should be $\hat{F} \approx F$. Now, from the structural layout we can conclude that the successive outputs of FFNN, Elman neural network and LRNNIFT structures are primarily dependent on the present and past values of the plant input-output values. The effectiveness of any estimation technique can be proven only if it minimises the requirement of plant knowledge. Therefore, in case of LRNNIFT, the proposed identification structure, we have taken only two inputs from the large set of plant variables-one past value, Y_p(k-1) and a past value of external input, r(k-1). The plant behaviour is predicted based on these two inputs only and is denoted by Y_{IFT}(k). The selection of fewer inputs (in this case, two) tones the structural and computational complexity by reducing the number of parameters that are to be updated.

Fig. 2 depicts the structure of suggested LRNNIFT model which is of series-parallel orientation having the below dynamical model:

$$V_{IFT}(k) = \hat{F}[y_q(k-1), r(k-1)] \tag{6}$$

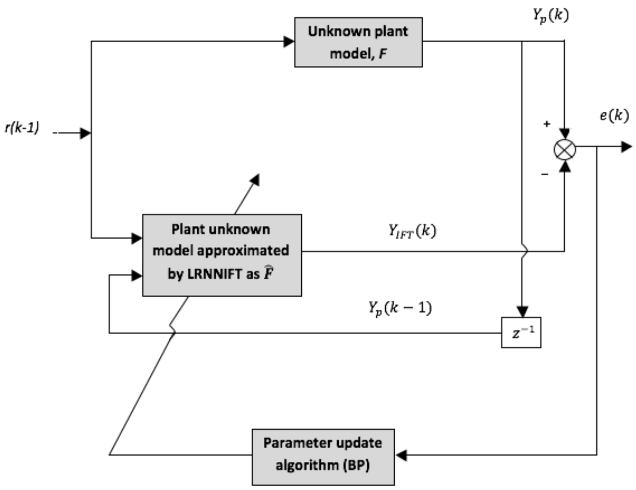


Fig. 2. Series-parallel identification model structure for LRNNIFT (Proposed).

B. Learning algorithm for LRNNIFT identification model

The standard back-propagation method is implemented for error reduction during training stage. It is based on the popular gradient-descent algorithm used for optimisation. The main idea behind the algorithm is to find the maximum or minimum (in this case, minimum) of a function. It uses the gradient at present value to iteratively compute the next value while minimising the function. Our aim is to generate an LRNNIFT based identifier which can estimate the plant's behaviour successfully. In order to attain this, we need to reduce the error (the gap between desired and actual output) generated during training. Conventionally, mean square error (MSE) function is used as the function to be minimised as it provides stable output compared to absolute error. The same is used below and expressed as:

$$E_i(k) = \frac{1}{2}[Y_p(k) - Y_{IFT}(k)]^2 = \frac{1}{2}e_i^2(k) \tag{7}$$

Here, E_i(k) represents the cost function for LRNNIFT training model and e_i(k) represents the error which is back propagated. The input weights, locally recurrent weights, feed through weights and the output weights are to be updated in this model. The parameter update expression of all these are presented subsequently below.

C. Tuning of Output and Feed-through Weights

The error gradient with respect to any j_{th} output layer weight can be expressed as

$$\frac{\partial E_i(k)}{\partial W_O^j(k)} = \left(\frac{\partial E_i(k)}{\partial Y_{IFT}(k)} \times \frac{\partial Y_{IFT}(k)}{\partial W_O^j(k)}\right) \tag{8}$$

where $\frac{\partial Y_{IFT}(k)}{\partial W_0^j(k)}$ = $T_j(k)$

Thus, each value in $W_0(k)$ = $w_0^1(k)$, $w_0^2(k)$,....$w_0^q(k)$ will be updated to a new value as following:

$$W_0^j(k) = W_0^j(k-1) - \Delta W_0^j(k-1) \tag{9}$$

where $\Delta W_0^j(k-1)$ = $-\eta e_i(k-1)T_j(k-1)$

In a similar manner for the feed through weights, each value in M= M_1, M_2,...M_q will be updated to a new value as following:

$$M_0^j(k) = M_0^j(k-1) - \Delta M_0^j(k-1) \tag{10}$$

where $\Delta M_0^j(k-1)$ = $-\eta e_i(k-1)T_j(k-1)$

D. Tuning of locally recurrent weights

The update equation for any j_{th} recurrent weight vector can be written as below:

$$\frac{\partial E_i(k)}{\partial W_F^j(k)} = \left(\frac{\partial E_i(k)}{\partial Y_{IFT}(k)} X \frac{\partial Y_{IFT}(k)}{\partial W_F^j(k)}\right) \tag{11}$$

where $\frac{\partial Y_{IFT}(k)}{\partial W_F^j(k)}$ = $\frac{\partial Y_{IFT}}{\partial V(k)} \frac{\partial V(k)}{\partial W_F^j(k)}$

Now $\frac{\partial Y_{IFT}}{\partial V(k)}$ = 1. Evaluation of $\frac{\partial V(k)}{\partial W_F^j(k))}$ is as below:

$$\frac{\partial V(k)}{\partial W_F^j(k)} = \frac{\partial V(k)}{\partial T_j(k)} \frac{\partial T_j(k)}{\partial W_F^j(k))} \tag{12}$$

where

$$\frac{\partial V(k)}{\partial T_j(k)} = W_0^j(k) \tag{13}$$

and

$$\frac{\partial T_j(k)}{\partial W_F^j(k)} = \left(1 - T_j^2(k)\right)\left[T_j(k-1) + W_F^j(k)L_j(k-1)\right] \tag{14}$$

where $L_j(k)$ = $\frac{\partial T_j(k)}{\partial W_T^j(k)}$ and $L_j(0)$ = 0. So, each element of local weight vector will be updated as:

$$W_T^j(k) = W_F^j(k-1) - \Delta W_F^j(k-1) \tag{15}$$

where

$$\begin{aligned}\Delta W_F^j(k) = -\eta e_i(k) W_0^j(k)[1 - T_j^2(k)][T_j(k-1) \\ + W_T^j(k)L_j(k-1)]\end{aligned} \tag{16}$$

E. Tuning of Input Weights

The update equation for input weight vector is also formulated using the above concept. For any j_{th} value, the equation can be given as:

$$W_I^j(k) = W_I^j(k-1) - \Delta W_I^j(k-1) \tag{17}$$

where

$$\Delta W_I^j(k) = -\eta e_i(k) W_0^j(k)[1 - T_j^2(k)][x_j(k) + W_F^j Q_j(k-1)] \tag{18}$$

where $Q_j(k) = \frac{\partial T_j(k)}{\partial W_I^j(k)}$ and $Q_j(0) = 0$

IV. Simulation Study

Post mathematical discussion, we now evaluate the efficiency of the proposed LRNNIFT based prediction strategy by implementing it on a plant and comparing the results with FFNN and ENN based prediction strategies. To conduct exhaustive analysis we have considered a complex dynamic plant. In the below example, we have taken only single hidden layer, 3 hidden neurons, equal learning rate and equal number of epochs in all the identifiers. The reason why we considered similar parameters for the proposed LRNNIFT based identification model and the other two models (ENN, FFNN) is to form a basis for comparison.

Consider a nonlinear time-delayed dynamical system whose dynamics are not known and is represented as mentioned below:

$$y_p(k) = F[x_1, x_2, x_3, x_4, x_5] \tag{19}$$

where $x_1 = y_p(k-1)$, x_2=$y_p(k-2)$, $x_3 = r(k-1)$, $x_4 = r(k-2)$, x_5 =$r(k-3)$ where the form of the unknown function is expressed as below:

$$y_p(k) = 0.72x_1 + 0.025x_2x_3 + 0.001x_4^2 + 0.2x_5 \tag{20}$$

The identification structure of FFNN, ENN and LRNNIFT is same and is given by following equation

$$Y_{IFT}(k) = \hat{F}[x_1, x_3] \tag{21}$$

In the above case, all the parameters of all three identifiers are trained by each epoch. The number of hidden neurons and the number of inputs for all three identifiers are 3 and 2, respectively. The total number input-output training data was 500 and the training was done for 500 epochs, post which it was graphically observed that the models were successfully trained. The learning rate (η) was set to constant value of 0.0025 for each variable of identifier. For the training, we have considered the external input r(k) to be a bounded sinusoidal signal with a range of [1, 1]. It is defined as $r(k) = sin(\frac{2\pi k}{250})$.

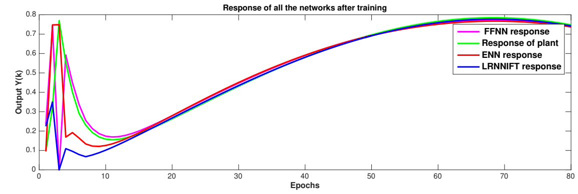


Fig. 3. Response of identifiers during initial stages of training

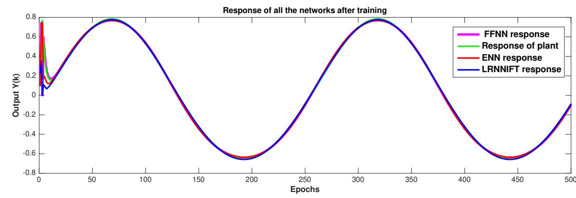


Fig. 4. Response of identifiers post training stage

A. Discussion on Simulation Results

Fig.3 and Fig.4 show the responses of identifiers during the initial and final stages of training. Evidently, we can conclude that the LRNNIFT has successfully predicted the plant behaviour better than the ENN and FFNN identifiers. Upon observing the MSE (Mean Square Error) and MAE (Mean Average Error) plots (shown in Figs.5-6) obtained during the training, we can again deduce that the LRNNIFT identifier has given the lowest value of error along-with fastest convergence of instantaneous error among all three identifiers. The numerical observations obtained from the MSE and

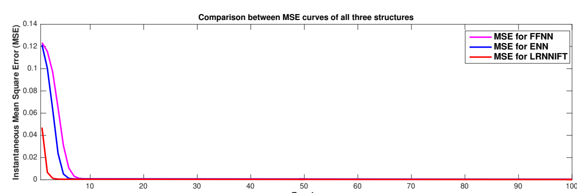


Fig. 5. Comparison of MSE curves of all three identifiers

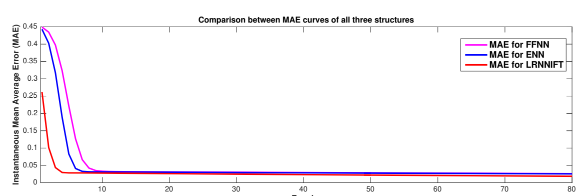


Fig. 6. Comparison of MAE curves of all three identifiers

MAE plots are given in form of comparison in Table 1. From the table, the average MSE (AMSE) and total MAE (TMAE) values are found to be the lowest in case of LRNNIFT identifier. We can also observe that with the same number of inputs, LRNNIFT model is performing better both in terms of error reduction and rate of convergence. This makes it more computationally efficient than FFNN and ENN.

TABLE I
DETAILED OUTPUT COMPARISON OF ENN, FFNN AND LRNNIFT

Parameters	Comparison Table		
	ENN	***FFNN***	***LRNNIFT***
Average MSE	8.615×10^{-5}	5.19×10^{-5}	1.5659×10^{-5}
Total MAE	0.011	0.0083	0.0047

B. Validation/Testing

Once the training of models is completed, the models are set to be tested with different inputs to the plant. This stage is important because it puts the quality of training on check and we can observe the actual efficiency of the identifier. In other words, this stage categorizes how well the plant is trained by the sample so that it can successfully predict output for data outside its sample space. Here, we take a different external input (having same range as that of the input used during training). The input subjected to plant is varied in nature to check the versatility of the plant's prediction capabilities.

The external input is given as follows:

If k is greater than 4 and less than or equal to 250,
$r(k) = sin\frac{\pi k}{25}$

If k is greater than 250 and less than or equal to 500,
$r(k) = 1$

If k is greater than 500 and less than or equal to 750,
$r(k) = -1$

If k is greater than 750,
$r(k) = 0.3sin\frac{\pi k}{25} + 0.1sin\frac{\pi k}{32} + 0.6sin\frac{\pi k}{10}$

The validation response plots of all three identifiers can be seen in Fig.7. It can be observed that all three identifiers have been trained successfully with LRNNIFT showing superior results.

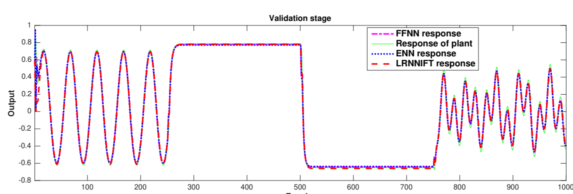


Fig. 7. Comparison of plant output with identifier responses during validation

V. CONCLUSION

In this paper, an alternative model is proposed for prediction of nonlinear systems with unknown dynamics. The LRNNIFT structure developed consists of locally recurrent loops along with input fed through weights directly to the output. The error back-propagation method is implemented for tuning of parameters and error reduction by optimising the cost function. The model is tested on a complex nonlinear system and compared with feed forward (FFNN) and Elman neural network (ENN) identification models. From the simulation study, it is observed that this model performs better in terms of learning dynamics for fewer inputs when compared to the other two reference models. Although the results of ENN and FFNN are comparable, despite the slightly high error in ENN model, we can observe that it has better convergence rate as compared to FFNN. Thus, LRNNIFT can be regarded as a general identification model that can be applied to wide class of nonlinear dynamic systems for the purpose of behavior prediction.

REFERENCES

[1] Deng, Jiamei. "Dynamic neural networks with hybrid structures for nonlinear system identification." Engineering Applications of Artificial Intelligence 26.1 (2013): 281-292.

[2] T. Wang, H. Gao and J. Qiu, "A Combined Adaptive Neural Network and Nonlinear Model Predictive Control for Multirate Networked Industrial Process Control," in IEEE Transactions on Neural Networks and Learning Systems, vol. 27, no. 2, pp. 416-425, Feb. 2016, doi: 10.1109/TNNLS.2015.2411671.

[3] Hägglund, Tore, and K. J. Åström. "REVISITING THE ZIEGLER-NICHOLS TUNING RULES FOR PI CONTROL—PART II THE FREQUENCY RESPONSE METHOD." Asian Journal of Control 6.4 (2004): 469-482.

[4] Hunt, Kenneth J., et al. "Neural networks for control systems—a survey." Automatica 28.6 (1992): 1083-1112.

[5] Isidori, Alberto, E. D. Sontag, and M. Thoma. Nonlinear control systems. Vol. 3. London: springer, 1995.

[6] Ding, Feng, and Tongwen Chen. "Identification of Hammerstein nonlinear ARMAX systems." Automatica 41.9 (2005): 1479-1489.

[7] T.-T. Lee, J.-T. Jeng, The Chebyshev-polynomials-based unified model neural networks for function approximation, IEEE Trans. Syst. Man Cybern. Part B (Cybern.) 28 (6) (1998) 925–935.

[8] Nelles, Oliver. Nonlinear System Identification: From Classical Approaches to Neural Networks, Fuzzy Models, and Gaussian Processes. Springer Nature, 2020.

[9] Savran, Aydogan, Ramazan Tasaltin, and Yasar Becerikli. "Intelligent adaptive nonlinear flight control for a high performance aircraft with neural networks." ISA transactions 45.2 (2006): 225-247

[10] Bloch, Gérard, and Thierry Denoeux. "Neural networks for process control and optimization: Two industrial applications." ISA transactions 42.1 (2003): 39-51.

[11] Akpan, Vincent A., and George D. Hassapis. "Nonlinear model identification and adaptive model predictive control using neural networks." ISA transactions 50.2 (2011): 177-194.

DELCON 2022

The maiden "IEEE Delhi Section International Conference on Electrical, Electronics and Computer Engineering (DELCON-2022)" is one of the finest International Conference covering broad topics in the areas of Electrical, Computer and Electronics Engineering, will be organized this year by Netaji Subhas University of Technology, Dwarka, New Delhi, India. DELCON conference will be organized annually at various locations in Delhi Section. This conference will provide an excellent platform to the researchers to present their research work and will be known as the Delhi-section's conference. The conference is technically and financially sponsored by IEEE Delhi Section. There are multiple tracks in the conference covering almost all areas of Electrical, Computer & Electronics Engineering.
All accepted & presented papers of the conference by duly registered authors will be submitted to IEEE Xplore for Publication (Conference Record Number: #54057). DELCON proceedings are indexed by SCOPUS and Google Scholar.

IEEE DELHI SECTION

Delhi Section is one of the 12 Sections in India coming under Asia-Pacific Region, the Region 10 of IEEE. At present it covers entire northern part of the country consisting of the four states of Rajasthan, Haryana, Punjab, Himachal Pradesh, and the National Capital Territory of Delhi, Union Territories of Chandigarh, Jammu & Kashmir, Ladakh. IEEE Delhi Section, which started on May 13, 1976 (after remaining as Sub-section since 1974).

MORE

Prashasti Srivastava ; Rajesh Kumar **All Authors**

Abstract:
This paper presents a locally recurrent neural network with input feed through (LRNNIFT) for the identification of nonlinear time-delayed system. The proposed network model parameters are tuned using a Back-propagation (BP) algorithm. The performance of the proposed model is compared with the well known recurrent Elman neural network (ENN) and a single layer feed-forward neural network (FFNN). The simulation results showed that the proposed model has shown better accuracy as compared to the other two models.

Published in: 2022 IEEE Delhi Section Conference (DELCON)

Date of Conference: 11-13 Feb. 2022

Date Added to IEEE *Xplore*: 20 April 2022

ISBN Information:

DOI: 10.1109/DELCON54057.2022.9753622

Publisher: IEEE

Conference Location: New Delhi, India

DELCON 2022

IEEE DELHI SECTION

NSUT

IEEE NSUT

CERTIFICATE

OF PRESENTATION

IS PROUDLY PRESENTED TO:

PRASHASTI SRIVASTAVA, Delhi Technological University
Paper title: A Novel Recurrent Neural Network Model for the Identification of Time-Delayed Non-Linear System

For their outstanding support in DELCON 2022, Organised by NETAJI SUBHAS UNIVERSITY OF TECHNOLOGY, NEW DELHI, INDIA HELD BETWEEN 11th-13th February 2022.

PROF.PRERNA GAUR

GENERAL CHAIR DELCON 2022

IEEE students

IEEE

IEEE Control Systems Society

IAS IEEE INDUSTRY APPLICATIONS SOCIETY Linking Research to Practice

TITAGARH

PPS International

PRASHASTISRIVASTAVA 2K20CI06 <prashastisrivastava_2k20ci06@dtu.ac.in>

SIGMA-2022 notification for paper 140

1 message

SIGMA-2022 <sigma2022@easychair.org> Tue, May 24, 2022 at 6:03 PM
To: Prashasti Srivastava <prashastisrivastava_2k20ci06@dtu.ac.in>

Dear Authors:

Congratulations! your Paper ID 140: Indirect Adaptive Control of Nonlinear system using Recurrent Neural Network, has been accepted for presentation in 2022 Second International Conference on Signals, Machines and Automation (SIGMA 2022) to be organized by Netaji Subhas University of Technology, New Delhi, India, India from August 5-6, 2022.

The following documents are to be sent as zip file (as Paper Id.zip) on the email(nsut.sigma2022@gmail.com)on or before 1st July, 2022 to complete the registration formalities:

1). Camera ready manuscript with the incorporated Reviewer comments.
2). The camera-ready manuscript should be in Springer conference template. The zip file and the paper should have the "paper id_paper title" as name (ex: 100_Review of FACTS Devices)
3). Filled copy right form available on conference website.
4). Scanned copy of the duly filled registration form and proof of registration fees paid. The registration amount and account details are given on conference website.

We are looking forward for your benign presence in SIGMA 2022.

Regards,

Conference Chair
SIGMA 2022
email: nsut.sigma2022@gmail.com
website: https://sites.google.com/view/sigma2022

SUBMISSION: 140
TITLE: Indirect Adaptive Control of Nonlinear system using Recurrent Neural Network

----------------------- REVIEW 1 ---------------------
SUBMISSION: 140
TITLE: Indirect Adaptive Control of Nonlinear system using Recurrent Neural Network
AUTHORS: Prashasti Srivastava and Rajesh Kumar

----------- Overall evaluation -----------
SCORE: 1 (weak accept)
----- TEXT:
What is the motivation of using the proposed neural network for the system considered ?
The system has been checked for a reference signal. The results should include the performance with disturbances also.

----------------------- REVIEW 2 ---------------------
SUBMISSION: 140
TITLE: Indirect Adaptive Control of Nonlinear system using Recurrent Neural Network
AUTHORS: Prashasti Srivastava and Rajesh Kumar

----------- Overall evaluation -----------
SCORE: 3 (strong accept)
----- TEXT:
The paper "Indirect Adaptive Control of Nonlinear system using Recurrent Neural Network" is well written. In this paper, authors have proposed a NN structure which acts as a controller for non-linear dynamical system.
They have also done a comparison between the proposed controller, FFNN and Elman based controller and proved

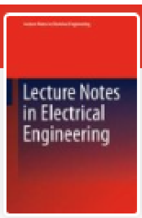
SIGMA 2022

About the Conference

The 2nd International Conference on Signals, Machines and Automation aims to bring together leading academicians and researchers to exchange and share their experiences and research outcomes within the spectrum of conference theme. SIGMA-2022 is going to be hosted by Netaji Subhas University of Technology (NSUT), New Delhi on 5th - 6th August 2022 in **virtual mode**. The Conference will provide excellent opportunities for the presentation of interesting new research results and discussion about them, leading to knowledge transfer and the generation of new ideas.

'It is planned to publish the peer reviewed and selected papers of conference as **proceedings with Springer** in their prestigious "**Lecture Notes in Electrical Engineering**" series (https://www.springer.com/series/7818). For detailed instructions for author and editors of conference proceedings, kindly visit the following link: https://www.springer.com/us/authors-editors/conference-proceedings. Select papers from the conference will be published by Springer as a proceedings book volume. Springer will conduct quality checks on the accepted papers and only papers that pass these checks will be published. Abstracts/extended abstracts and short **papers (less than 4 pages) are not considered for publication.**'

Extended version of some of the selected papers will also be invited for publication in ***"International Journal of Nonlinear Control and Dynamics"*** **–Inderscience**.



Book series

Lecture Notes in Electrical Engineering

Editorial board

About this book series

The book series *Lecture Notes in Electrical Engineering* (LNEE) publishes the latest developments in Electrical Engineering - quickly, informally and in high quality. While original research reported in proceedings and monographs has traditionally formed the core of LNEE, we also encourage authors to submit books devoted to supporting student education and professional training in the various fields and applications areas of electrical engineering. The series cover classical and emerging topics concerning: — show all

Electronic ISSN
1876-1119

Print ISSN
1876-1100

Abstracted and indexed in

DBLP
EI Compendex
INSPEC
Japanese Science and Technology Agency (JST)
SCImago
SCOPUS
WTI Frankfurt eG
zbMATH

Springer

PRASHASTI SRIVASTAVA 2K20CI06 M.Tech C&I Thesis

by Prashasti Srivastava

Submission date: 30-May-2022 12:41PM (UTC+0530)
Submission ID: 1847028446
File name: PlagueCheck2.docx (1.17M)
Word count: 7817
Character count: 42611

PRASHASTI SRIVASTAVA 2K20CI06 M.Tech C&I Thesis

ORIGINALITY REPORT

18%	14%	9%	%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	kupdf.net Internet Source	1%
2	www.elprocus.com Internet Source	1%
3	Www.tutorialspoint.com Internet Source	1%
4	Rajesh Kumar, Smriti Srivastava, J. R. P. Gupta. "Modeling and adaptive control of nonlinear dynamical systems using radial basis function network", Soft Computing, 2016 Publication	1%
5	doczz.net Internet Source	1%
6	Rajesh Kumar, Smriti Srivastava, J.R.P. Gupta, Amit Mohindru. "Diagonal recurrent neural network based identification of nonlinear dynamical systems with Lyapunov stability based adaptive learning rates", Neurocomputing, 2018 Publication	1%

7	manualzz.com Internet Source	1%
8	www.bcgsc.ca Internet Source	1%
9	paulhmuller.files.wordpress.com Internet Source	1%
10	ftp.cngb.org Internet Source	1%
11	idoc.pub Internet Source	<1%
12	brage.bibsys.no Internet Source	<1%
13	Roberto Bonfigli, Stefano Squartini. "Chapter 3 Background", Springer Science and Business Media LLC, 2020 Publication	<1%
14	tesi.cab.unipd.it Internet Source	<1%
15	hdl.handle.net Internet Source	<1%
16	www.bx.psu.edu Internet Source	<1%
17	www.encyclopedias.biz Internet Source	<1%

18	link.springer.com Internet Source	<1 %
19	www.bombjack.org Internet Source	<1 %
20	Donald W. K. Andrews. "Testing When a Parameter is on the Boundary of the Maintained Hypothesis", Econometrica, 2001 Publication	<1 %
21	arxiv.org Internet Source	<1 %
22	Rajesh Kumar, Smriti Srivastava, J.R.P. Gupta. "Diagonal recurrent neural network based adaptive control of nonlinear dynamical systems using lyapunov stability criterion", ISA Transactions, 2017 Publication	<1 %
23	upcommons.upc.edu Internet Source	<1 %
24	www.science.gov Internet Source	<1 %
25	e2a.nits.ac.in Internet Source	<1 %
26	S. Palani. "Chapter 2 Continuous and Discrete Time Systems", Springer Science and Business Media LLC, 2022 Publication	<1 %

27	ijarcs.info Internet Source	<1 %
28	pubs.aina.ucalgary.ca Internet Source	<1 %
29	Xu, LiXin, Zhao Yang Dong, and Arthur Tay. "TIME SERIES FORECAST WITH ELMAN NEURAL NETWORKS AND GENETIC ALGORITHMS", Advances In Natural Computation, 2004. Publication	<1 %
30	www.ijitee.org Internet Source	<1 %
31	光 西片, 卓也 倉沢, 峯晴 杉本, 徹 赤坂 et al. "真菌アレルギーにおけるRASTの臨床的有用性の検討", アレルギー, 1988 Publication	<1 %
32	mnit.ac.in Internet Source	<1 %
33	publications.lib.chalmers.se Internet Source	<1 %
34	www.hindawi.com Internet Source	<1 %
35	Rajesh Kumar, Smriti Srivastava, J.R.P Gupta, Amit Mohindru. "Temporally local recurrent radial basis function network for modeling	<1 %

	and adaptive control of nonlinear systems", ISA Transactions, 2019 Publication	
36	de.mathworks.com Internet Source	<1 %
37	jurnal.unmer.ac.id Internet Source	<1 %
38	mdpi.com Internet Source	<1 %
39	"Proceedings of the International Conference on Advanced Intelligent Systems and Informatics 2021", Springer Science and Business Media LLC, 2022 Publication	<1 %
40	128.104.165.254 Internet Source	<1 %
41	Fan, Guo-Feng, Li-Ling Peng, Wei-Chiang Hong, and Fan Sun. "Electric load forecasting by the SVR model with differential empirical mode decomposition and auto regression", Neurocomputing, 2016. Publication	<1 %
42	MISRA, B. B., P. K. DASH, and G. PANDA. "OPTIMIZED POLYNOMIAL FUZZY SWARM NET FOR CLASSIFICATION", Advances in Computer Science and Engineering Texts, 2010.	<1 %

Publication

43	S. Palani. "Signals and Systems", Springer Science and Business Media LLC, 2022 Publication	<1 %
44	Sanjit Kumar Dash, Satyam Raj, Rahul Agarwal, Jibitesh Mishra. "Automobile Predictive Maintenance Using Deep Learning", International Journal of Artificial Intelligence and Machine Learning, 2021 Publication	<1 %
45	Ying Han, Anguo Zhang, Qing Chen, Wei Zhu. "Improved Integrate-and-Fire Neuron Models for Inference Acceleration of Spiking Neural Networks", Institute of Electrical and Electronics Engineers (IEEE), 2019 Publication	<1 %
46	Yuri Flores Albuquerque. "Shape optimization and level set method applied to inversion problems", Universidade de Sao Paulo, Agencia USP de Gestao da Informacao Academica (AGUIA), 2021 Publication	<1 %
47	espace.curtin.edu.au Internet Source	<1 %
48	ethesis.nitrkl.ac.in Internet Source	<1 %
49	hal.univ-lorraine.fr	

Internet Source

<1%

50	uwe-repository.worktribe.com Internet Source	<1%
51	www.osapublishing.org Internet Source	<1%
52	"Introduction to Discrete Event Systems", Springer Science and Business Media LLC, 2008 Publication	<1%
53	Guanghua Ren, Yuting Cao, Shiping Wen, Tingwen Huang, Zhigang Zeng. "A modified Elman neural network with a new learning rate scheme", Neurocomputing, 2018 Publication	<1%
54	Howard J. Shaughnessy, Frances Friewer, Adele Snyder. "Comparative Efficiency of Rectal Swabs and Fecal Specimens in Detecting Typhoid and Salmonella Cases and Carriers", American Journal of Public Health and the Nations Health, 1948 Publication	<1%
55	Lecture Notes in Computer Science, 2001. Publication	<1%