Turkish Journal of Electrical Engineering & Computer Sciences

http://journals.tubitak.gov.tr/elektrik/

Research Article

Turk J Elec Eng & Comp Sci
(2013) 21: 1151 – 1165

doi:10.3906/elk-1107-12

Comparison study of computational parameter values between LRN and NARX in identifying nonlinear systems

Farah Hani NORDIN,* Farrukh Hafiz NAGI, Aidil Azwin ZAINUL ABIDIN
Centre for Signal Processing and Control Systems, College of Engineering, Universiti Tenaga Nasional, Jalan IKRAM-UNITEN, 43000 Kajang, Selangor, Malaysia

Received: 07.07.2011 • **Accepted:** 21.11.2011 • **Published Online:** 03.06.2013 • **Printed:** 24.06.2013

Abstract: To determine the nonlinear autoregressive model with exogenous inputs (NARX) parameter values is not an easy task, even though NARX is reported to successfully identify nonlinear systems. Apart from the activation functions, number of layers, layer size, learning rate, and number of epochs, the number of delays at the input and at the feedback loop need to also be determined. The layer recurrent network (LRN) is seen to have the potential to outperform NARX. However, not many papers have reported on using the LRN to identify nonlinear systems. Therefore, it is the aim of this paper to investigate and analyze the parametric evaluation of the LRN and NARX in identifying 3 different types of nonlinear systems. From the 3 nonlinear systems, the satellite's attitude state space is more complex compared to the sigmoid and polynomial equations. To ensure an unbiased comparison, a general guideline is used to select the parameter values in an organized manner. The LRN and NARX performance is analyzed based on the training and architecture parameters, mean squared errors, and correlation coefficient values. The results show that the LRN outperformed NARX in training quality, needs equal or fewer parameters that need to be determined through heuristic processes and equal or lower number of epochs, and produced a smaller training error compared to NARX, especially when identifying the satellite's attitude. This indicates that the LRN has the capability of identifying a more complex and nonlinear system compared to NARX.

Key words: Layer recurrent network, nonlinear autoregressive with exogenous inputs, nonlinear system identification, recurrent neural network

1. Introduction

Networks with recurrent connections have been known to have important capabilities not found in feedforward networks, where recurrent connections allow information about events occurring at arbitrary times in the past to be retained and used in current computations, which also allows networks to generate complex behaviors [1]. The commonly used recurrent network is the nonlinear autoregressive model with exogenous inputs (NARX). The successful applications of the NARX series-parallel architecture have been reported in many papers [2–4], especially in the area of predictive control, while the NARX parallel architecture was also reported to successfully identify many nonlinear systems [5–10].

Even though NARX is reported to successfully identify nonlinear systems, Liutkevičius [11] mentioned in his paper that NARX is impractical for the modeling of high-level dynamic processes. When NARX is integrated with an intelligent system, such as a fuzzy system, it gives a promising approach, but the disadvantage is that

*Correspondence: farah@uniten.edu.my

the rule-based generation will become complex and often impossible because of the lack of necessary knowledge, especially for highly nonlinear complex systems [12]. Furthermore, to select the value of the NARX parameters is not an easy task. Apart from the unavoidable parameters, such as the activation functions, number of layers, layer size, learning rate, and number of epochs that need to be determined through heuristic process, due to the NARX architecture, the number of delays at the input and at the feedback loop need to also to be determined. Lin et al. [13] used intelligent memory order selection through a pruning process to determine the number of delays used in the NARX. However, a good initial heuristic is still needed to significantly improve the performance.

The layer recurrent network (LRN) is a current network that was developed from the basic and earlier simplified version introduced by Elman [14]. The LRN generalizes the Elman network by not only having an arbitrary number of layers and arbitrary transfer functions in each layer, but it also trains the LRN using exact versions of the gradient-based algorithms of standard backpropagation [15].

The LRN and NARX are both dynamic recurrent neural networks, but the LRN has feedback loop(s) at every layer, except the output layer, and does not contain any delay at the input vector. The NARX parallel architecture, on the other hand, has delay(s) at the input and a feedback connection from the output of the output layer to the input layer.

Theoretically, the LRN should be able to generalize better than NARX since the LRN has recurrent connections, which allow the network's hidden units to see its own previous output; thus, subsequent behavior can be shaped by previous responses [14]. However, only a few researchers [16,17] have reported using the LRN to identify nonlinear systems. This may be because there are already a variety of neural network model structures to choose from and it is human nature to start with the simplest and most familiar architectures, such as multilayer perceptron and radial basis function, and then move on to a more complex architecture, such as NARX, if the feedforward network architecture could not give satisfactory results.

It is difficult to decide on a neural network model, especially when identifying a nonlinear system. To try with all of the available models and see which model suits the best is not a good idea since it can take months or years to come out with a final model. The question that needs to be asked is if a network that can model a complex and nonlinear system with minimum training parameters and training epochs exists, then why is that network not given priority when deciding on a model in identifying a nonlinear system?

In this paper, the parametric evaluation and the performance between the LRN and NARX are compared based on the identification of 3 nonlinear systems with different types of nonlinearities, which are addressed in Section 2. Section 3 describes the training parameters, while Section 4 presents the training process. The results and discussion are explained in Section 5, and the conclusion is presented in Section 6.

2. Nonlinear systems

Three nonlinear systems with different types of nonlinearities are identified: a) the nonlinear satellite's attitude state space model, which is made up by trigonometric functions; b) the sigmoid function, which is made up by an exponential function of a nonlinear equation; and c) a nonlinear equation, which is made up by a power of polynomial equation. The satellite's attitude state space model is more complex (in terms of the mathematical model) compared to the sigmoid and polynomial equation.

2.1. Trigonometric functions: satellite's attitude state space model

The satellite's attitude was chosen as one of the systems to be identified due to its complex nonlinear attributes. The satellite's attitude is controlled by 3 rotational angles, also known as Euler angles, about the satellite's body axes as follows: roll $(\emptyset)$, about the X_B axis; pitch (θ), about the Y_B axis; and yaw (ψ), about the Z_B axis. Assuming that the satellite is a rigid satellite with no moving elements inside of it, the attitude dynamic equation is obtained from the well-known Euler's moment equation [18], where the rotational motion of a body caused by the applied moment is examined. Eq. (1) is the Euler's moment equation:

$$M = \dot{h}_B + [\omega_B \times h_B], \tag{1}$$

where **M** is the applied moment (from the thruster), **h** is the angular momentum vector of a rigid body, and ω is the angular velocity, while the subscript **B** indicates a derivative in the rotating *body* frame. Rearranging Eq. (1) gives:

$$\dot{h}_B = M - [\omega_B \times h_B]. \tag{2}$$

The law of angular momentum [19] allows for expressing the equations of motion in terms of angular velocity, as in Eq. (3):

$$h = I \cdot \omega, \tag{3}$$

where $\boldsymbol{I}$ is the inertia axes, defined as:

$$I = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix}. \tag{4}$$

Substituting Eq. (3) into Eq. (2) gives:

$$\begin{aligned} I \cdot \dot{\omega_B} &= M - [\omega_B \times I \cdot \omega_B], \\ \dot{\omega_B} &= I^{-1} \cdot M - \left[I^{-1} \cdot \omega_B\right] \times \left[I \cdot \omega_B\right]. \end{aligned} \tag{5}$$

In general, the body angular velocity relative to the reference frame is denoted by:

$$\omega_B = \left[p_B, q_B, r_B\right]^T, \tag{6}$$

and **M** is the applied moment to the satellite, which is defined as:

$$M = \left[M_x, M_y, M_z\right]^T. \tag{7}$$

Substituting Eqs. (4), (6), and (7) into Eq. (5) leads to Eq. (8):

$$\begin{bmatrix} \dot{p}_B \\ \dot{q}_B \\ \dot{r}_B \end{bmatrix} = \begin{bmatrix} \frac{M_x - q_B \cdot I_{zz} \cdot r_B + r_B \cdot I_{yy} \cdot q_B}{I_{xx}} \\ \frac{M_y - r_B \cdot I_{xx} \cdot p_B + p_B \cdot I_{zz} \cdot r_B}{I_{yy}} \\ \frac{M_z - p_B \cdot I_{xx} \cdot r_B + q_B \cdot I_{yy} \cdot p_B}{I_{zz}} \end{bmatrix}. \tag{8}$$

Take note that Eq. (8) must first be integrated in order to have the body angular velocity in terms of the applied moments, as shown in Eq. (9):

$$\begin{bmatrix} p_B \\ q_B \\ r_B \end{bmatrix} = \omega_B = \int \begin{bmatrix} \dot{p_B} \\ \dot{q}_B \\ \dot{r}_B \end{bmatrix}. \tag{9}$$

The relationship between the body-fixed angular velocity vector, $[p_B, q_B, r_B]^T$, and the rate of change of the Euler angles, $\left[\dot{\emptyset}, \dot{\theta}, \dot{\psi}\right]^T$, can be determined by resolving the Euler angle rates into the body-fixed coordinate frame [18]. The transformation from the body angular velocity to the Euler angle rates with respect to the reference axes, X_R , Y_R, and Z_R, is done by the following transformation:

$$\begin{bmatrix} \dot{\emptyset} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \frac{\sin(\emptyset)\sin(\theta)}{\cos(\theta)} & \frac{\cos(\emptyset)\sin(\theta)}{\cos(\theta)} \\ 0 & \cos(\emptyset) & -\sin(\emptyset) \\ 0 & \frac{\sin(\emptyset)}{\cos(\theta)} & \frac{\cos(\emptyset)}{\cos(\theta)} \end{bmatrix} \cdot \begin{bmatrix} p_B \\ q_B \\ r_B \end{bmatrix}, \tag{10}$$

where $\begin{bmatrix} 1 & \frac{\sin(\emptyset)\sin(\theta)}{\cos(\theta)} & \frac{\cos(\emptyset)\sin(\theta)}{\cos(\theta)} \\ 0 & \cos(\emptyset) & -\sin(\emptyset) \\ 0 & \frac{\sin(\emptyset)}{\cos(\theta)} & \frac{\cos(\emptyset)}{\cos(\theta)} \end{bmatrix}$ is the simplified direction cosine matrix [A] [18].

Substituting Eq. (8) and Eq. (9) into Eq. (10) produces Eq. (11), which is the Euler angle rates in terms of the applied moments.

$$\begin{bmatrix} \dot{\emptyset} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \frac{\sin(\emptyset)\sin(\theta)}{\cos(\theta)} & \frac{\cos(\emptyset)\sin(\theta)}{\cos(\theta)} \\ 0 & \cos(\emptyset) & -\sin(\emptyset) \\ 0 & \frac{\sin(\emptyset)}{\cos(\theta)} & \frac{\cos(\emptyset)}{\cos(\theta)} \end{bmatrix} \cdot \int \begin{bmatrix} \dot{p}_B \\ \dot{q}_B \\ \dot{r}_B \end{bmatrix} \tag{11}$$

Integrating the above equation, the Euler angles, roll $(\emptyset)$ pitch (θ), and yaw (ψ), are obtained, by which the body's frame is rotated relative to the reference frame with the applied moments, $\mathbf{M}$, injected to the satellite as its input.

The applied moments, $\mathbf{M}$, are obtained from the thrusters used in the attitude control. The thrusters are activated by a pulsing mode that is generated by applying 3 random signals with Gaussian distribution, which are then independently transformed, using a relay, into a 2-level signal. A random signal with Gaussian distribution is used because a random signal contains random frequencies and amplitudes. With a random range of frequencies and amplitudes, it is expected to capture most of the characteristics or dynamics of the nonlinear system.

The 3 random signals with Gaussian distribution are generated using 3 'Random Source' Simulink blocks, with the mean and variance set to 0 and 1, respectively. The seed of the random signal is set to 'repeatable', where the block randomly selects an initial seed once and uses the same seed every time simulation starts. Since the block randomly selects the initial seed, the random signals between the 3 blocks are different, and since the same seed is used every time the simulation starts, this ensures consistency in every simulation. The random signal is generated on a discrete mode with a switching time equal to 2 s, which was observed to be able to stabilize the system's response and was sampled at 0.5 s, fulfilling the Nyquist theorem. The random signal is then converted into a 2-level signal, using a relay, between the reaction thruster torque levels. The reaction torque levels, ranging between 0.01 Nm and 30 Nm, are very common in most spacecraft [18]. However, in this paper, the thruster torque level is set to 0.281 Nm, following the satellite data in [20].

The relay threshold applied to the Gaussian signal causes the input to be nonlinear, where for highly negative values of the input the output will be at the lower level of the negative value and for highly positive values of the input the output will be at the upper level of the positive value. In this paper, the relay lower

level of the negative value is set to –1 and the upper level of the positive value is set to +1. The relay lower level of the negative and upper level of the positive values are the thruster torque level, where any change in the attitude is taken care of by firing the thruster. Since the thruster torque level is set to 0.281 Nm, this causes the lower and upper values of the signal to be –0.281 Nm and +0.281 Nm, respectively. This is consistent with Figure 1, where the lower and upper level values of the relay are multiplied with the thruster torque level value.

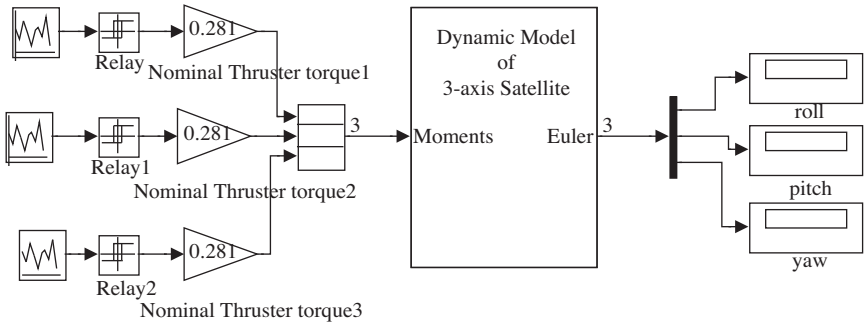


Figure 1. Set up of an open loop satellite's attitude state space numerical experiment.

In an ideal case, the switching or firing of the thruster should start when the input is less than or greater than zero. In reality, there will be a delay between the time output changes sign and the switching time. This delay is also known as chattering, where in this paper, it is set to ±10 m rad, following the data given in [20]. The characteristic of the relays used is shown in Figure 2, with the chattering value set to ±10 m rad.

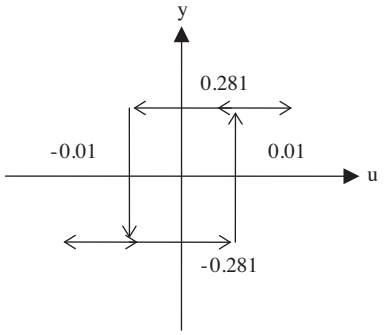


Figure 2. Relay characteristic.

Once the 3 sets of the 2-level signals are generated, they are concatenated to form a 3-dimension of a 2-level input signal, which acts as the applied moments, **M**, and the input to the satellite's attitude state space model.

2.2. Exponential function: sigmoid function of a nonlinear equation

The following nonlinear system [21] is used as a second model to be identified by the LRN and NARX:

$$y(n) = \frac{1 - e^{-g(n)}}{1 + e^{-g(n)}}, \tag{12}$$

where $g(n)$ is a function of:

$$g(n) = 2.2y(n-1) + 1.77y(n-2) - 0.52y(n-3) + u(n) - 1.25u(n-1) + 0.65u(n-2) - 0.35u(n-3) \tag{13}$$

The system $y(n)$ is a sigmoid function that is made up by an exponential function of a nonlinear function, $g(n)$. The nonlinear function depends on the current input and past inputs and outputs, as shown in Eq. (13).

2.3. Power of polynomial equation: nonlinear equation

The third system that the LRN and NARX identify is a power of polynomial equation [22] defined as:

$$y(n+1) = 0.15y^2(n) + 0.3y(n-1) + 0.6u^3(n) + 0.18u^2(n) - 0.24u(n), \tag{14}$$

where the system's future output does not only depend on its current and past outputs, but also on its current input. The input in Eq. (14) is defined as:

$$u(n) = 0.7\sin(\frac{2\pi n}{60}). \tag{15}$$

3. Neural network training parameters

The LRN and NARX parameters include the number of hidden layers and neurons, activation function, learning rate, number of delays, and number of epochs, which need to be determined during training. The following describes how each of these parameters is determined.

3.1. Number of hidden layers and neurons

In this paper, the neural network is initially trained using the basic architecture of the neural network, i.e. 1 hidden layer and 1 output layer, with 1 neuron in each layer. A network with multiple hidden layers is more prone to getting caught up in undesirable local minima [23]. Since Duda et al. [23] and Villiers and Barnard [24] stated that 3 layers are sufficient to implement any arbitrary function, this paper will restrict the network's maximum number of layers to 3.

The number of neurons represents the number of output states of a layer. For the case of the satellite's attitude state space model, which has 3 output states, a decoupling method is used where 1 axis is not affecting the other axis. Therefore, only 1 neuron is needed at the output layer because only 1 output is observed from a network. The number of neurons in a hidden layer is increased gradually until an optimum number of neurons is reached. The optimum number of neurons refers to the number of neurons before the training performance starts to decrease.

3.2. Activation functions

Since the LRN and NARX use backpropagation, they need activation functions that can calculate their own derivative, as presented by Eqs. (16) through (18):

$$\text{Hyperbolic tangent sigmoid} : \varphi(n) = \tanh(n), \tag{16}$$

$$\text{Logistic sigmoid} : \varphi(n) = \frac{1}{1+e^{-n}}, \tag{17}$$

$$\text{Pure linear} : \varphi(n) = n. \tag{18}$$

In this paper, the hidden layer(s) are initially set to the hyperbolic tangent sigmoid function, which is an antisymmetric function with respect to its origin, because according to Haykin [25], an antisymmetric function

learns faster (in terms of training iterations) than when it is nonsymmetric. An activation function, $\varphi(v)$, is an antisymmetric function if $\varphi(-v) = -\varphi(v)$, which is not satisfied by the standard logistic sigmoid function [25]. The output layer is fixed to the pure linear function so that the output generated can be mapped to any values from $-\infty$ to $+\infty$.

3.3. Number of epochs

Kermani et al. [26] presented in their paper that as the number of epochs is increased, the mean squared error remains flat for a while and then goes through a rapid decay before it flattens out again, which is an indication that no further learning is taking place and the network's memorization is beginning. Therefore, the number of epochs needs to be increased gradually so that the network does not go in memorization state. In this paper, the network is initially trained with 10 epochs and this is then increased by 10 epochs per training.

3.4. Learning rate

The selection of the learning rate is a trade-off between training the network with more epochs and the network accuracy. Even though accuracy plays an important role, to train a network with bigger epoch is not practical since a longer time is needed for training and the network is pruned for overtraining. In this paper, the learning rate is set between 0 and 1 (inclusive). The initial learning rate value is set to 0.01 and is increased or decreased by a factor of 0.01, depending on the error obtained and on the number of epochs required to train the network.

3.5. Number of delays

Currently, there is no proper method for determining the number of delays of a network, other than through heuristic processes. In this paper, the number of delays is initially set to 1 at the feedback loop and 0 at the input (for NARX). A minimum delay of 1 is needed at the feedback loop to ensure the stability of the network. The training starts with initializing the parameter value and the training error. Since in real cases, it is almost impossible to get zero error, the target error is set to 0.01 when identifying the satellite's Euler angles responses. For the identification of the sigmoid function of a nonlinear equation and the power of polynomial equation, the target errors are set to 0.001.

4. Training process

To ensure that an unbiased comparison is made, based on the discussion made in Section 3, the following steps are used to determine the 5 parameter values and train the LRN and NARX in an organized and systematic approach.

Step 1: Set the targeted training error and the initial or minimum values of the network parameters.

Step 2: Train and simulate the network. If the error is more than the targeted training error, the parameter value is changed one at a time before it is retrained and resimulated in the following manner:

Step I: Increase the number of neurons. If there is an improvement in the training error, the number of neurons is further increased until no further significant improvement on the error is obtained.

Step II: Change the activation function of the hidden layer(s) from hyperbolic tangent sigmoid to logistic sigmoid or pure linear function and increase the number of neurons. If the error is greater than the target error, proceed with the activation function that gives the minimum error with the minimum number of neurons.

Step III: Add 1 hidden layer to the network and repeat Step I and II. The maximum number of layers is 3. If the training error is still greater than the target error, proceed to the next step.

Step IV: Increase the number of epochs by 10 (with the learning rate fixed at the initial learning rate value, i.e. 0.01). If the error does not decrease significantly, proceed to Step V.

Step V: Increase the learning rate by a factor of 0.01. If the error decreases, Step IV is repeated. If increasing the number of epochs does not reduce the error significantly, then the learning rate is decreased by a factor of 0.01 and Step IV is repeated.

Step VI: Increase the number of delays.

Figure 3 shows the pseudocode for defining and training the LRN architecture for the Euler angles of the satellite's attitude state space model. Note that a similar pseudocode was also used to define and train the roll and yaw responses of the satellite's attitude state space model, the sigmoid function of nonlinear equation, and the power of polynomial equation.

```
Load input data;
Load target/output data;
Convert the format of the input from concurrent (matrix) to sequential (cell);
Convert the format of the output from concurrent (matrix) to sequential (cell);
Define the LRN architecture;
Define the learning rate value;
Define the number of epochs;
Initialize the LRN;
Train the LRN;
Simulate the LRN;
Convert the simulated response from sequential (cell) to concurrent (matrix);
Error = target response – simulated response;
Calculate the mean squared error;
```

Figure 3. Pseudocode for defining, training, and simulating the LRN architecture.

The NARX series-parallel and parallel architectures are designed and trained based on the pseudocode shown in Figure 4. A similar pseudocode is used to define and train the roll and yaw responses of the satellite's attitude state space model, the sigmoid function of nonlinear equation, and the power of polynomial equation.

5. Results and discussion

The quality of the network is assessed based on the performance of the network in reproducing the measured data, where the trained LRN and NARX are transformed into Simulink blocks and the respective input is applied concurrently to the respective plants and the trained neural network models.

Figure 5a shows the Euler angles of the satellite's attitude state space model and the LRN responses, where it is clear that the Euler angles produced by the LRN models are duplicating the Euler angles produced by the satellite's attitude state space model. Referring to Table 1, the mean squared error for the roll and pitch responses of the LRN are very small, at 0.0145 rad and 0.000204 rad, respectively. The small errors are consistent with Figure 5a, where the LRN models duplicate the plant responses. The Euler angles of the

```
Load input data;
Load target/output data;
Convert the format of the input from concurrent (matrix) to sequential (cell);
Convert the format of the output from concurrent (matrix) to sequential (cell);
Define the number memory order;
Define the input and target/output data in parallel and delayed by: memory order +1;
Define the target/output data and delayed by: memory order +1;
Define the connection of the input delay;
Define the number of output delays;
Define the NARX series-parallel architecture;
Define the size of the input;
Define the learning rate value;
Define the number of epochs;
Initialize the NARX series-parallel;
Initialize the NARX parallel (for at least 2 of the redefining process);
Train the NARX series-parallel;
Convert from series-parallel to the NARX parallel architecture;
Define the input and delayed by: memory order +1;
Simulate the NARX parallel;
Convert simulated response from cell to matrix format;
Error = target – simulated response;
Calculate the mean squared error;
```

Figure 4. Pseudocode for defining, training, and simulating the NARX architecture.

satellite's attitude state space model and the NARX responses are shown in Figure 5b. For the roll, the NARX response is able to follow the pattern of the plant response, except between 95 and 110 s. This explains the mean squared error of 2.1994 rad obtained during training. The pitch response of the NARX model is able to follow the pattern of the plant response, which reflects the small training error produced by the NARX model, which is 0.2836 rad. The yaw response of the NARX model is also able to follow the transient of the state space response, except between 20 and 102 s, which contributed to the 2751.2 rad of error obtained during training. To further validate the LRN and NARX models, the correlation coefficient, ρ, of the LRN with the target responses and the correlation coefficient of the NARX with the target responses are computed and are shown in the respective figures.

The outputs from the plant and the LRN of the sigmoid function of a nonlinear equation are shown in Figure 6a, where it can be seen that the LRN response closely follows the plant's response after about 2.5 s and the responses between the LRN and the plant cannot be distinguished after about 5.0 s. The responses from

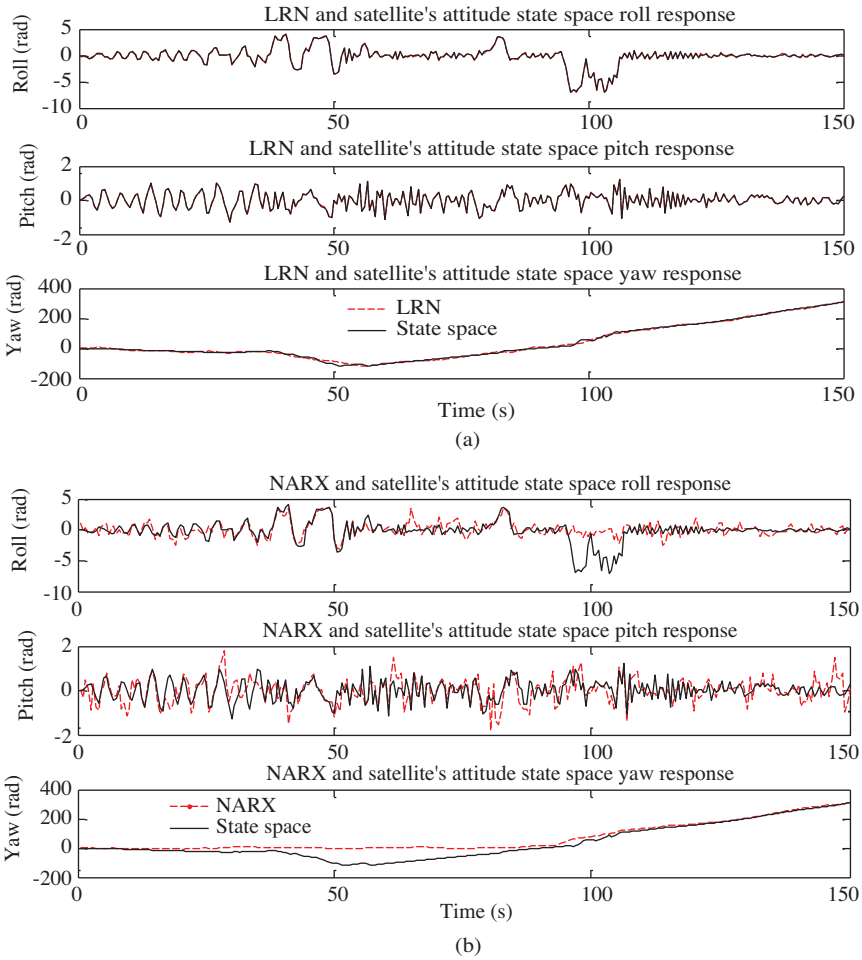


Figure 5. (a) Euler angle responses of the LRN and the satellite's attitude state space model. (b) Euler angle responses of the NARX and the satellite's attitude state space model.

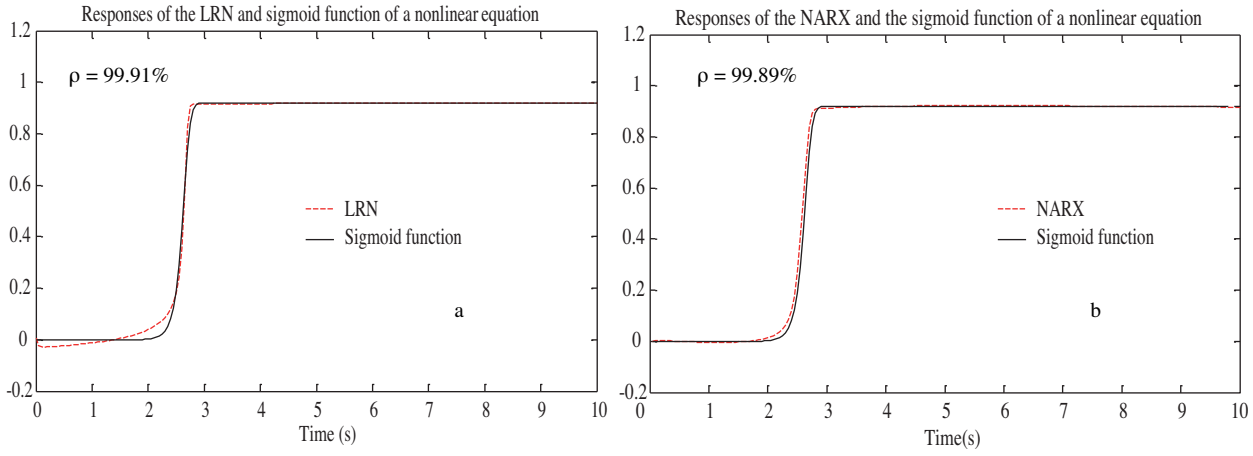


Figure 6. (a) Responses of the LRN and sigmoid function of a nonlinear equation. (b) Responses of the NARX and sigmoid function of a nonlinear equation.

the plant and the NARX model are shown in Figure 6b, where it can be seen that the NARX response also closely follows the plant's response. However, the NARX took more than 5.0 s to reach the steady state error.

The outputs from the plant and the LRN model are shown in Figure 7a, where it can be seen that response produced by the LRN closely follows the response produced by the polynomial equation. The responses from the plant and the NARX model are shown in Figure 7b, where the NARX response also very closely follows the plant's response and consistence with a smaller error produced compared to the LRN.

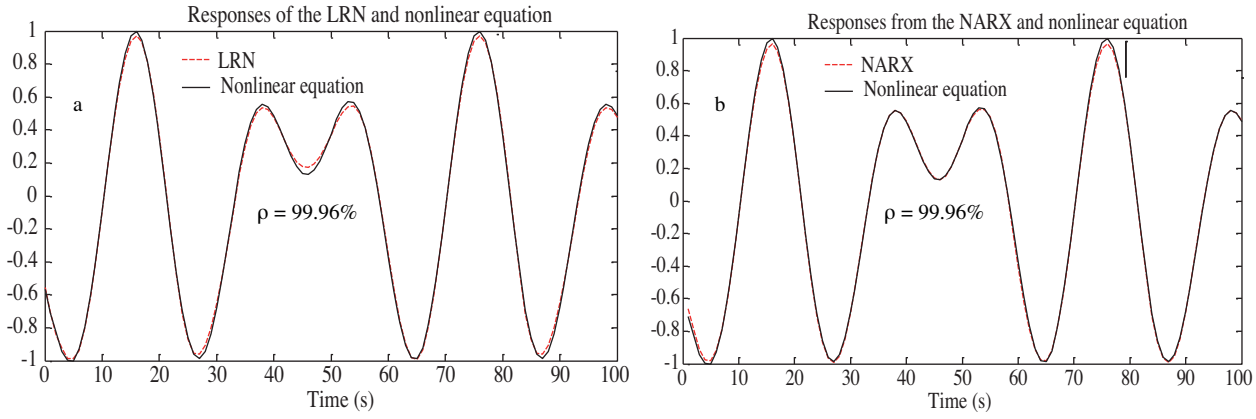


Figure 7. (a) Responses of the LRN and the power of polynomial equation. (b) Responses of the NARX and the power of polynomial equation.

The LRN correlation coefficient in identifying the satellite's attitude is, on average, 32.24% higher than that of the NARX model. The correlation coefficient produced by the LRN model in identifying the sigmoid function of a nonlinear equation is also 0.02% higher than that of the NARX model. However, the correlation coefficient obtained by the LRN model is the same as that of the NARX model when the power of polynomial equation was identified. Since a higher correlation coefficient value reflects a higher model quality, it can be concluded that the quality between the LRN and NARX models are the same for the power of polynomial equation, but the LRN models performed better when identifying the satellite's attitude state space model and the sigmoid function of the nonlinear equation compared to the NARX models.

An additional analysis of the correlation of the residuals is made and it is shown that most of the dynamics of the nonlinear system have been captured by the LRN model. Figure 8 shows the results of the autocorrelation residual test of the LRN in identifying the dynamics of the satellite's attitude state space model. The cross-correlation between the input and output residual is expected to be zero since there is no correlation between the output residual and the input residual.

The training mean squared error and the number of parameters that need to be determined are concluded in Table 1 and the details of the parameter values are shown in Table 2. Table 2 lists the 5 network parameter values for the LRN and NARX models in identifying the 3 nonlinear systems. Referring to Tables 1 and 2, the LRN model outperformed the NARX model in identifying the satellite's attitude state space model, where the LRN only needs 3 out of 5 parameters that need to be determined through heuristic processes with 200 epochs and produced a training error 536.52 times smaller than that of the NARX model.

Although both the LRN and NARX models only require the layer size to be determined through heuristic processes in identifying the sigmoid function of a nonlinear equation, the LRN model still outperformed the

NARX because the LRN model needs 2 neurons less than the NARX model to produce a mean squared error of 8.004e-5 times smaller than that of the NARX model.

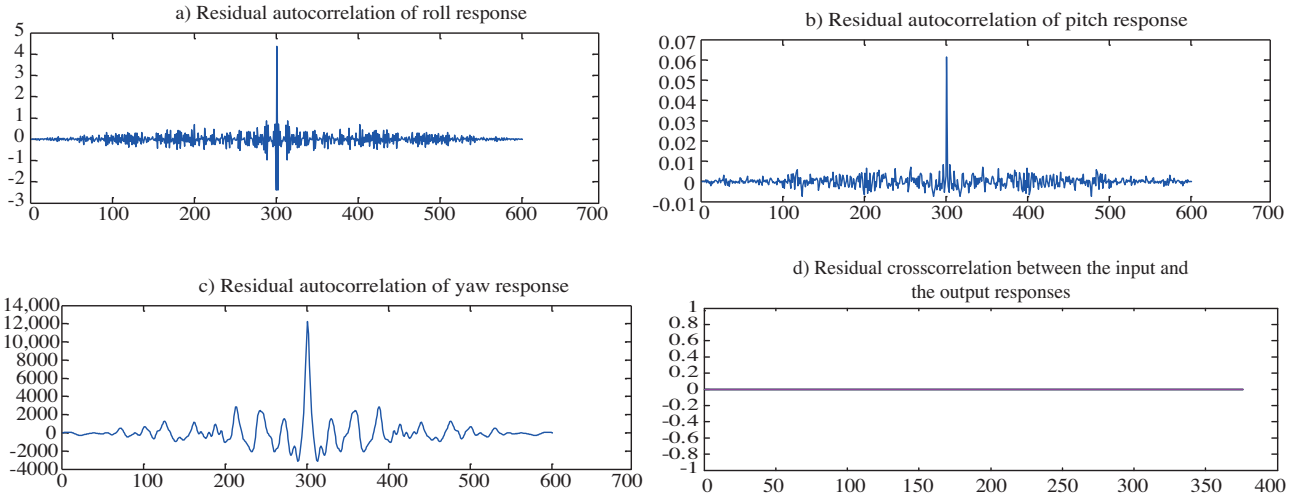


Figure 8. Correlation residual test.

Table 1. The mean squared error and number of parameter values achieved by the LRN and NARX models in identifying the nonlinear systems.

Criteria	Nonlinear models	LRN		NARX
Mean squared error	Satellite's attitude	Roll	0.0145	2.1994
		Pitch	0.000204	0.2836
		Yaw	40.6562	2751.2
	Sigmoid function	2.8104e-4		3.6108e-4
	Polynomial equation	3.7967e-4		2.3109e-4
No. of parameters determined heuristically	Satellite's attitude	Roll	3 parameters	4 parameters
		Pitch	3 parameters	5 parameters
		Yaw	3 parameters	5 parameters
	Sigmoid function	1 parameter		1 parameter
	Polynomial equation	1 parameter		1 parameter

The NARX model outperformed the LRN in the training performance when identifying the power of polynomial equation. Even though the parameter values for both the LRN and NARX models are the same and only the layer size needs to be determined through heuristic processes, the error produced by the NARX model is 1.4858e-4 times smaller than that of the LRN model.

Therefore, it can be concluded that the LRN outperformed the NARX model in training quality when identifying the satellite's attitude state space model and the sigmoid function of the nonlinear equation. The LRN requires equal or fewer parameters that need to be determined through heuristic processes and an equal or lower number of epochs, and also produced a smaller training error compared to the NARX model. Even though the NARX outperformed the LRN in terms of the training error when identifying the power of polynomial equation, both models require only the layer size to be determined through heuristic processes and require the same value of the layer size.

Table 2. Parameter values set for the LRN and NARX models in identifying the nonlinear systems.

Parameters	Nonlinear models	LRN		NARX
No. of epochs	Satellite's attitude	Roll	200 epochs	1500 epochs
		Pitch	200 epochs	1000 epochs
		Yaw	200 epochs	1500 epochs
	Sigmoid function	10 epochs*		10 epochs*
	Polynomial equation	10 epochs*		10 epochs*
No. of layers	Satellite's attitude	Roll	3 layers	2 layers*
		Pitch	3 layers	3 layers
		Yaw	3 layers	3 layers
	Sigmoid function	2 layers*		2 layers*
	Polynomial equation	2 layers*		2 layers*
Layer size (no. of neurons in each layer)	Satellite's attitude	Roll	[10 10 1]	[15 1]
		Pitch	[10 10 1]	[10 10 1]
		Yaw	[10 3 1]	[7 10 1]
	Sigmoid function	[2 1]		[4 1]
	Polynomial equation	[5 1]		[5 1]
Learning rate	Satellite's attitude	Roll	0.01*	0.05
		Pitch	0.01*	0.0001
		Yaw	0.01*	0.5
	Sigmoid function	0.01*		0.01*
	Polynomial equation	0.01*		0.01*
No. of delays	Satellite's attitude	Roll	1*	4
		Pitch	1*	5
		Yaw	1*	2
	Sigmoid function	1*		1*
	Polynomial equation	1*		1*

*Initial value.

6. Conclusion

In this paper, a comparison between LRN and NARX model performances was made based on 3 training and architecture parameter criteria in identifying 3 different types of nonlinear systems. The first criterion was to reduce the number of parameters, such as the number of layers, neurons, delays, epochs, and learning rates, that are selected heuristically. The second criterion was to compare the parameter values, where a minimum value is preferred to speed up the training process. The third criterion was to compare the mean squared error and correlation coefficient obtained, where it reflects the training quality. From the analysis of this paper, it is demonstrated that the LRN model not only requires the minimum parameters to be determined heuristically and has the ability to speed up the training process, but it also produces equal or better model quality compared to the NARX model, especially in identifying a more complex and nonlinear system. Thus, the LRN should be given priority when identifying a complex and nonlinear system.

References

[1] H. Lee, Y. Park, K. Mehrotra, C. Mohan, S. Ranka, "Nonlinear system identification using recurrent networks", IEEE International Joint Conference on Neural Networks, Vol. 3, pp. 2410–2415, 1991.

[2] D. Psichogios, L. Ungar, "Nonlinear internal model control and model predictive control using neural networks", Proceedings of the 5th IEEE International Symposium on Intelligent Control, pp. 1082–1087, 1990.

[3] R. Gao, A. O'Dwyer, E. Coyle, "Model predictive control of CSTR based on local model networks", Proceedings of the Irish Signals and Systems Conference, pp. 397–402, 2002.

[4] D. Soloway, P.J. Haley, "Neural generalized predictive control: A Newton-Raphson implementation", Proceedings of the 11th IEEE International Symposium on Intelligent Control, pp. 277–282, 1996.

[5] M. Basso, L. Giarre, S. Groppi, G. Zappa, "NARX models of an industrial power plant gas turbine", IEEE Transactions on Control Systems Technology, Vol. 13, pp. 599–604, 2005.

[6] H. Luo, S. Puthusserypady, "NARX neural networks for dynamical modelling of fMRI data", Proceedings of the International Joint Conference on Neural Networks, pp. 542–546, 2006.

[7] I. Maio, I. Stievano, F.G. Canavero, "NARX approach to black-box modelling of circuit elements", Proceedings of the IEEE International Symposium on Circuits and Systems, Vol. 3, pp. 411–414, 1998.

[8] H.P.H. Anh, N.H. Phuc, "Inverse neural MIMO NARX model identification of nonlinear system optimized with PSO", Proceedings of the 5th IEEE International Symposium on Electronic Design, Test and Applications, pp. 144–149, 2010.

[9] H.P.H. Anh, "Inverse dynamic model identification of 2-axes PAM robot arm using neural MIMO NARX model", IEEE/ASME International Conference on Advanced Intelligent Mechatronics, pp. 1282–1287, 2009.

[10] E. Pisoni, M. Farina, C. Carnevale, L. Piroddi, "Forecasting peak air pollution levels using NARX models", Engineering Applications of Artificial Intelligence, Vol. 22, pp. 593–602, 2009.

[11] R. Liutkevičius, "Fuzzy Hammerstein model of nonlinear plant", Nonlinear Analysis: Modelling and Control, Vol. 13, pp. 201–212, 2008.

[12] K.K. Ahn, H.P.H. Anh, "Inverse double NARX fuzzy modeling for system identification", IEEE/ASME Transactions on Mechatronics, Vol. 15, pp. 136–148, 2010.

[13] T.N. Lin, C. Giles, B.G. Horne, S.Y. Kung, "A delay damage model selection algorithm for NARX neural networks", IEEE Transactions on Signal Processing, Vol. 45, pp. 2719–2730, 1997.

[14] J.L. Elman, "Finding structure in time", Cognitive Science, Vol. 14, pp. 179–211, 1990.

[15] H. Demuth, M. Beale, M. Hagan, Neural Network's Toolbox 5 User's Guide, MathWorks, 2007.

[16] M. Saggar, T. Mericli, S. Andoni, R. Miikkulainen, "System identification for the Hodgkin-Huxley model using artificial neural networks", International Joint Conference on Neural Networks, pp. 2239–2244, 2007.

[17] F.H. Nordin, F.H. Nagi, "Layer-recurrent network in identifying a nonlinear system", International Conference on Control, Automation and Systems, pp. 387–391, 2008.

[18] M.J. Sidi, Spacecraft Dynamics & Control: A Practical Engineering Approach, Cambridge, Cambridge University Press, 1997.

[19] F.C. Moon, Applied Dynamics: With Applications to Multibody and Mechatronic Systems, New York, Wiley, 1998.

[20] S. Thongchet, S. Kuntanapreeda, "A fuzzy-neural bang-bang controller for satellite attitude control", Journal of King Mongkut's Institute of Technology North Bangkok, Vol. 11, pp. 11–17, 2001.

[21] W. Gang, L. Hua, K.F. Man, S. Kwong, "A robust nonlinear system identification algorithm based on homotopy theory", IEEE International Conference on Systems, Man, and Cybernetics, Computational Cybernetics and Simulation, Vol. 2, pp. 1313–1316, 1997.

[22] X. Li, T. Chen, "System identification using a new dynamic neural network", Proceedings of International Joint Conference on Neural Networks, Vol. 3, pp. 2363–2366, 1993.

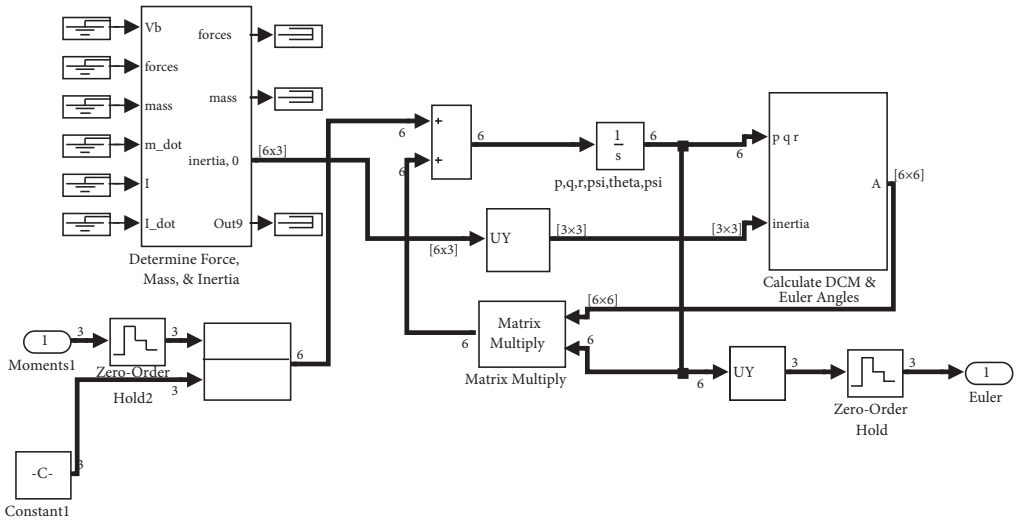
[23] R.O. Duda, P.E. Hart, D.G. Stork, Pattern Classification, New York, Wiley, 2001.

[24] J.D. Villiers, E. Barnard, "Backpropagation neural nets with one and two hidden layers", IEEE Transactions on Neural Networks, Vol. 4, pp. 136–141, 1992.

[25] S. Haykin, Neural Networks: A Comprehensive Foundation, New Jersey, Prentice Hall, 1999.

[26] B.G. Kermani, M.W. White, H.T. Nagle, "A new method in obtaining a better generalization in artificial neural networks", Proceedings of the 16th Annual International Conference of the IEEE Engineering in Medicine and Biology Society: Engineering Advances: New Opportunities for Biomedical Engineers, pp. 1119–1120, 1994.

Appendix



Block diagram of the 'dynamic model of the 3-axis satellite' block.