Deep Neural Network System Identification for Servomechanism System

MOHAMED. A. SHAMSELDIN
Mechanical Engineering Department, Future University in Egypt, Cairo, EGYPT

Abstract: This paper presents a systematic technique for designing the input signal to identify the one-stage servomechanism system. Sources of nonlinearities such as friction and backlash consider an obstacle to obtaining an accurate model. Also, most such systems suffer from a lack of system parameters data. So, this study establishes a model using the black-box modeling approach; simulations are performed based on real-time data collected by LabVIEW software and processed using MATLAB System Identification toolbox. The input signal for the servomechanism system driver is a pseudo-random binary sequence that considers violent excitation in the frequency interval and the output signal is the corresponding stage speed measured by rotary encoder. The candidate models were obtained using linear least squares, nonlinear least squares, and Deep Neural Network (DNN). The validation results proved that the identified model based on DNN has the smallest mean square errors compared to other candidate models.

Keywords: *Deep Neural Network (DNN), Harmony Search (HS), Servomechanism*
Received: June 28, 2021. Revised: March 19, 2022. Accepted: April 16, 2022. Published: May 11, 2022.

1. Introduction

The newest growth of machine tools is to realize high-speed spindle and feed drives, which causes improvement in the performance and reduces the machining cycle times. Besides, the design of feed drives with satisfying dynamic response and high performance now is considered necessary in several industrial applications.

Most of the real engineering systems are nonlinear systems but, the nonlinearity characteristics change according to the complexity degree of the system. It is known that the mechanical systems have high nonlinear behavior and parameters uncertainty because of the friction phenomenon and the backlash of gear trains and mechanical parts. For example, the one-stage servomechanism system which considers a base unit of CNC machines. So, the traditional mathematical model cannot achieve accurate presentation for this type of system. Experimental identification is a well-recognized methodology to obtain a precise process model, often intended for control but also for other purposes. The nature of the input signal for the identification has a great effect on the accuracy of the model. But, in many cases, the input signal cannot be easily selected with respect to plant behavior constrictions. Pseudo-Random Binary Sequences (PRBS) are often used as violent excitation signal for system identification, it has a finite length that can be synthesized frequently with simple generators while presenting favorable spectra for identification.

The family of candidate models for system identification can be classified along with several different aspects. The linear and nonlinear theory has been well developed and investigated to real applications through recent years. The linear structure is used to simplify the analysis where the parameters are constant and do not vary throughout a simulation, such as autoregressive with exogenous inputs (ARX) model. In contrast, a non-linear model presents dependent parameters that are permitted to vary throughout the period of a simulation run, and its use becomes necessary where interdependencies between parameters cannot be considered insignificant.

Artificial neural networks (ANN) trained by back-propagating error derivatives have the possibility to develop much better models of data that has nonlinear behavior. In recent researches, improvements in both computer hardware and machine learning algorithms have led to more effective approaches for training deep neural networks (DNNs) that contain several layers of non-linear hidden units and a very large output layer known as the deep learning algorithms.

Lately, deep learning has been become attractive and has significant attention from a wide range of engineering applications. Compare to the traditional neural networks, the vital features of deep learning are to have more hidden layers and neurons and to improve learning performance.

Using these features, complex and large problems that could not be solved with traditional neural networks can be resolved by deep learning algorithms. Therefore, deep learning has been subjected to various applications including pattern recognition and classification problems; for example, handwritten digit recognition [4], speech recognition [3], human action recognition [5], and so on. However, to the best knowledge of the authors, no result has been published in the system identification and automatic control field. Thus, this paper focuses on presenting the applying possibility of deep learning in system identification areas.

This paper investigates several candidate models for one stage servomechanism system based on linear and nonlinear least squares and deep neural networks. The actual input/output data is collected by NI Data Acquisition (DAQ) Card 6009 using LABVIEW software. The input signal is a Pseudo-Random Binary Sequences (PRBS) while the output signal is the corresponding stage speed measured by the optical encoder. Comparison between the candidate models is implemented to select the best between them.

The paper is ordered as follows: Section II illustrates the system modeling. The system identification techniques is involved in Section III. Section IV shows the validation results. Lately, Section V presents the conclusion.

2. Mathematical Model

It is well-known that several servomechanism system parameters cannot be determined easily. Consequently, most of the researchers neglect those parameters which will cause errors in the design of their own controllers. The structure of the lead-screw-driven stage is shown in Fig.1. The stage is stayed by linear guides on a bed. The driving mechanism contains a DC motor, coupling, screw shaft, and nut. The DC motor rotates the screw shaft while the nut changes the movement from rotation to linear motion and drives the stage. The stage speed and position are measured through a rotary encoder coupled with a screw shaft.

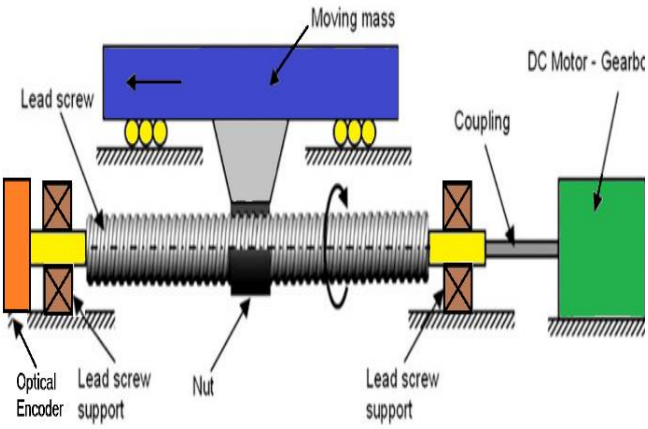

Fig. 1. The structure of the lead screw drive system.

A four-inertia model of a lead screw driving system is shown in Fig. 2. This model considers each component as a lumped constant, and the four moments of inertia of the components are connected by stiffness and damping parameters. The four inertia parameters are the mass of the screw shaft (M_b), the mass of the stage (M_t), the moment of inertia of the rotor (J_m), and the moment of inertia of the screw shaft (J_b), The moment of inertia of the rotor and that of the screw shaft (which are connected by the torsional stiffness of the coupling) is rotated by the motor torque, where θ_m and θ_b represent a motor angle and screw-shaft angle respectively. The motion of the screw shaft in the x-direction is taken into account, and the mass of the screw shaft has stayed on the axial stiffness of the screw shaft and its bearing, where X_b represents the gap between the screw shaft and nut (backlash). The parameters details in the four-inertia model are defined in Table I.

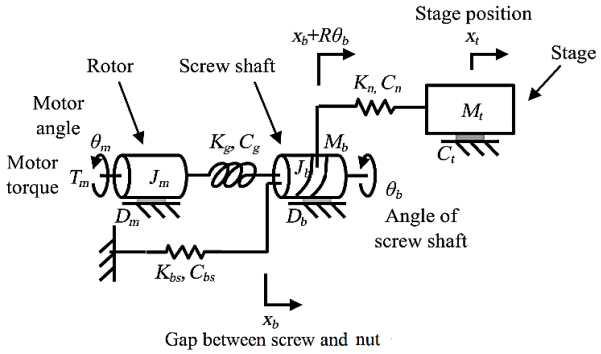

Fig. 2. Four-inertia model of lead screw driven stage.

The linear motion of the stage is generated by rotating the screw shaft through the nut, and $X_b + R\theta_b$ stands for the position at which the nut is mounted. The R is a transformation factor of the lead screw. The mass of the stage and the nut mounting position are linked through the axial stiffness of the nut, and X_t denotes the stage position. The overall system parameters are verified in table 1.

Table 1. Model Parameters

No.	Parameter	Unit	Content
1	J_m	kg·m2	Moment inertia of DC motor rotor
2	J_b	kg·m2	Moment inertia of screw shaft
3	M_b	kg	Mass of screw shaft
4	M_t	kg	Mass of Stage
5	K_g	N·m / rad	Torsional stiffness of screw shaft and coupling
6	K_{bs}	N / m	Axial stiffness of screw shaft and bearing
7	K_n	N / m	Axial stiffness of nut
8	D_m	N·m / (rad / s)	Rotational viscous damping of the motor
9	D_b	N·m / (rad / s)	Rotational viscous damping of screw shaft and bearing
10	C_t	N / (m / s)	Axial viscous damping of stage
11	R	m / rad	Conversion factor of lead screw
12	C_g	N·m / (rad / s)	Rotational viscous damping of screw shaft and coupling
13	C_{bs}	N / (m / s)	Axial viscous damping of screw shaft and bearing
14	C_n	N / (m / s)	Axial viscous damping of nut
15	θ_m	rad	Motor shaft angle
16	θ_b	rad	Screw shaft angle
17	X_b	m	Gap between screw shaft and nut (backlash)
18	X_t	m	Stage position
19	T_m	N.m	The DC motor torque

The model differential equations can be described as follows:

$$J_m.\ddot{\theta}_m + D_m.\dot{\theta}_m + C_g.(\dot{\theta}_m - \dot{\theta}_b) + K_g.(\theta_m - \theta_b) = T_m \quad (1)$$

$$J_b.\ddot{\theta}_b + D_b.\dot{\theta}_b + R.[C_n(R.\dot{\theta}_b + \dot{X}_b - \dot{X}_t) + K_n.(R.\theta_b + X_b - X_t)] = C_g.(\dot{\theta}_m - \dot{\theta}_b) + K_g.(\theta_m - \theta_b) \quad (2)$$

$$M_b.\ddot{X}_b + C_{bs}.\dot{X}_b + K_{bs}.X_b + C_n.(R.\dot{\theta}_b + \dot{X}_b - \dot{X}_t) + K_n.(R.\theta_b + X_b - X_t) = 0 \quad (3)$$

$$M_t.\ddot{X}_t + C_t.\dot{X}_t = C_n.(R.\dot{\theta}_b + \dot{X}_b - \dot{X}_t) + K_n.(R.\theta_b + X_b - X_t) \quad (4)$$

3. System Identification

3.1 Collecting Input/Output Data

This section demonstrates the main parts of the one-stage servomechanism system. Also, it shows the open-loop performance of the servomechanism system. Moreover, it demonstrates the system identification techniques and the validation results. Fig. 3 illustrates the main components of one stage servomechanism experimental setup which consists of a DC Motor Electro-Mechanical Module. The stroke of the stage ranges from 0 to 9 Inches. The DC motor has a nominal speed of 1800 rev/min, and an armature voltage of 90 V dc. The Optical Encoder is an add-on that provides position feedback signals (100 pulses per revolution). Two magnetic limit switches detect when the sliding block reaches the start or end position. The DC Motor Drive controls the DC Motor Electro-Mechanical Module. This versatile drive also allows an external signal to control the motor speed. A data acquisition card (DAQ) NI USB-6009 is connected to the computer that is used to perform the control algorithms.

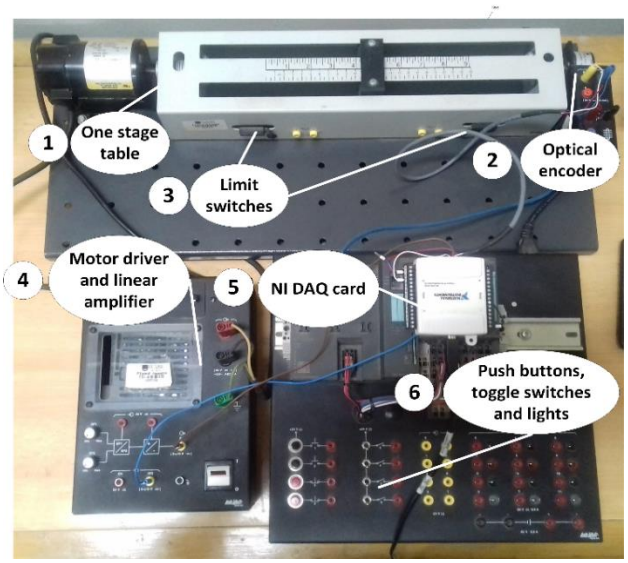

Fig. 3. The one Stage Servomechanism Experimental Setup.

The main idea of the program has been designed to make the NI DAQ 6009 generates an analog output signal (-5V to 5 V) to the linear amplifier. Also, the analog output signal from the optical encoder has been collected at the same time. The speed of the DC motor will fluctuate when the generated signal change continuously. The positive signal will cause the DC motor speed to fluctuate in the forward direction, while the DC motor will fluctuate in the reverse direction through the negative voltage ranges. The shaft of the optical encoder is coupled with the lead screw shaft to measure the speed and position of the stage as shown in Fig. 4.

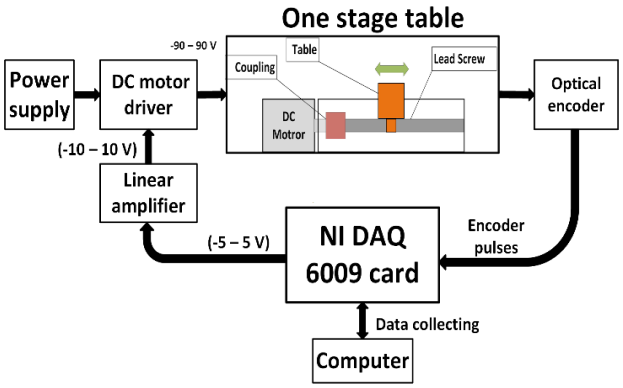

Fig. 4. The block diagram of experimental setup servomechanism system.

Fig. 5 demonstrates The PRBS output signal of the NI DAQ card and input to the DC motor driver while Fig. 6 shows the corresponding linear speed of the stage measured by the optical encoder.

Also, Fig. 7 displays the actual position of one stage servomechanism through the experiment.

It can be noted that the stage position increases in positive ranges of the input signal entering to DC motor driver while the actual stage position decreases in negative ranges of the input signal. The input /output data will be collected and stored in an excel sheet file and then this data will be used to develop identified model for the experimental setup.

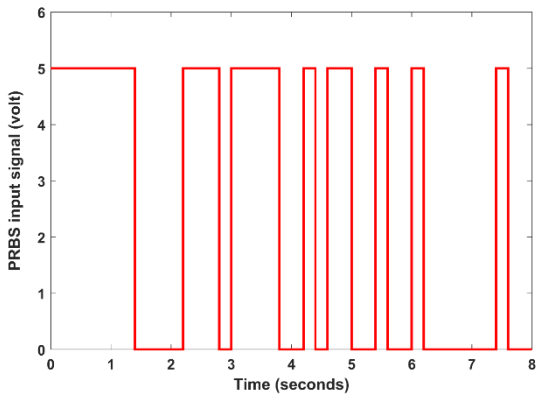

Fig. 5. The PRBS output signal of NI DAQ card and input to DC motor driver.

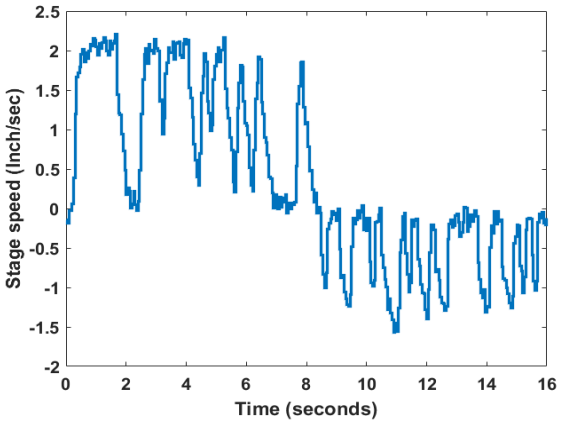

Fig. 6. The output linear speed of the servomechanism table.

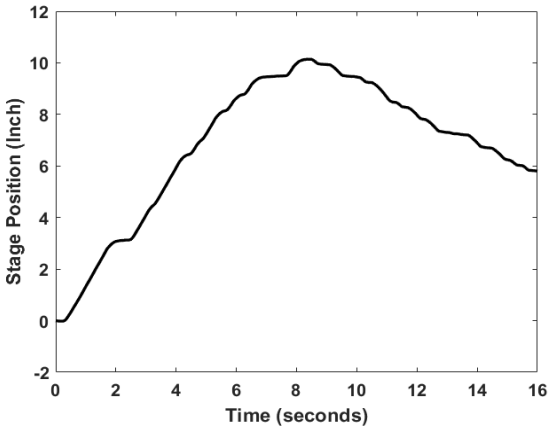

Fig. 7. The corresponding position of the servomechanism table.

The general linear transfer function can be expressed as follows:

$$\frac{y(s)}{u(s)} = \frac{a_k.S^k + a_{n-1}.S^{k-1} + \cdots + a_0}{b_n.S^n + b_{n-1}.S^{n-1} + \cdots + b_0} \quad (5)$$

In equation (5), $a_k, \ldots, a_0, b_n, \ldots, b_0$ are the estimated parameters of the approximate transfer function (T.F). The y(s) stands for linear stage speed while u(s) is the input voltage to DC motor drive.

The accuracy of the transfer function improves significantly when the system degree is increased. However, often there is a restriction that increasing order cannot make the model accuracy sufficiently [25]. Therefore, it is necessary to explicitly add the nonlinearities into the system.

As the servomechanism systems suffer from uncertainty and complex nonlinear dynamics, the nonlinear ARX (NLARX) model structure has been developed to model such systems [26].

The NLARX model consists of several complicated nonlinear functions which can model the backlash and friction in servomechanism systems significantly.

An NLARX model can be understood as an extension of a linear model. A linear SISO ARX model has this structure [27].

$$y(t) + a_1 y(t-1) + a_2 y(t-2) + \cdots + a_{na} y(t-na) = b_1 u(t) + b_2 u(t-1) + \cdots + b_{nb} u(t-nb+1) + e(t) \quad (6)$$

where u is input, y is output, na is the number of past output terms and nb is the number of past input terms.

This structure can be extended to create a nonlinear form where instead of the weighted sum that represents a linear mapping, the NLARX model has a more flexible nonlinear mapping function [28].

$$y(t) = f(y(t-1), y(t-2), y(t-3) \ldots., u(t), u(t-1), u(t-2), \ldots) \quad (7)$$

where f is a nonlinear function (to simulate the behavior of friction and backlash existing in servomechanism systems). Inputs to f are model regressors as shown in the following Fig. 8.

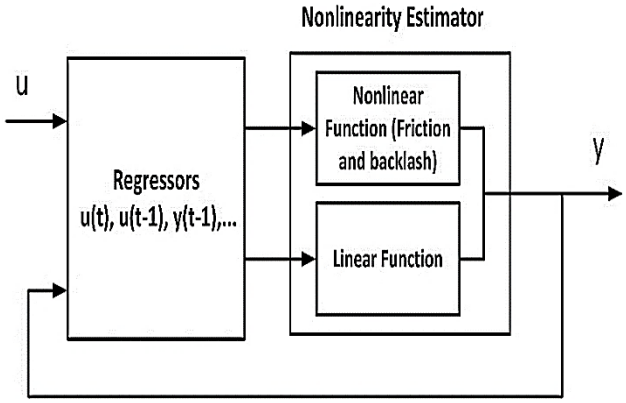

Fig. 8. The NLARX model structure [28].

The obtained input /output data was used to develop a set of linear and nonlinear identified models to select the best between them. This task can be carried using the MATLAB System Identification Toolbox. The input u is the input voltage to the DC motor driver. The output y contains the corresponding output linear velocity of the stage. The sampling interval is 50 milliseconds. Two models had been obtained, the first identified model is a linear second-order system while the second identified model is an NLARX model.

The linear second-order system has the following specifications.

$$\frac{y(s)}{u(s)} = \frac{2.108s + 0.02192}{s^2 + 5.691s + 0.06781} \quad (8)$$

By matching with equation (1), the estimated parameters of the linear approximation transfer function are $k = 1,\ n = 2, a_1 = 2.108\ , a_0 = 0.02192, b_2 = 1,\ b_1 = 13.91\ and\ b_0 = 78.45.$

On the other hand, the obtained discrete-time ARX model has the following structure.

$$A(z)y(t) = B(z)u(t) + e(t) \quad (9)$$

where $A(z) = 1 - 0.7659\,z^{-1} - 0.03264\,z^{-2}$
$B(z) = 0.03467\,z^{-1} + 0.05299\,z^{-2}$

By corresponding with equation (2), the polynomial orders are $na = 2$ and $nb = 2$. Also, the used nonlinear function is a wavelet with 25 units.

Artificial Neural Networks (ANN) had been proposed in 1943. The ANN has been used to solve different types of problems such as robotic processing, object recognition, speech and handwriting recognition, and even real-time sign-language translation. Despite the intuition, that deeper architectures would yield better results than the then more commonly used shallow ones, experimental tests with deep networks had found similar or even worse results when compared to networks with only one or two layers. The Latest deep architectures use several modules that are trained separately and stacked together so that the output of the first one is the input of the next one. There are two main types of ANN, the first type is the shallow neural network and the second type is the deep neural network. Deep learning has several layers of hidden units and it also permits many more factors to be used before over-fitting occurs. Thus, for deep learning, a deep architecture is used as shown in Fig. 9.

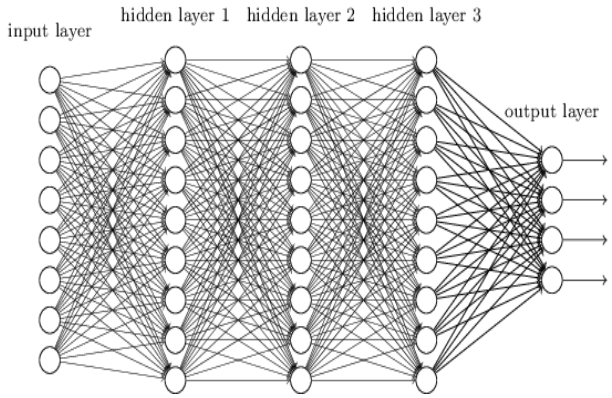

Fig. 9. Deep Neural Network.

The neural network structure in this study contains three layers the first layer is the input layer which receives the input PRBS signal to the DC motor driver. The second layer is the hidden layer that contains several hidden neurons and receives data from the input layer. The third layer is the output layer which presents the corresponding linear stage speed.

$x_i = (x_1, x_2, x_3, \dots, x_n)^t$ Input vector applied to the layer, the whole of the hidden neuron input 'j' is:

$$net_j^h = \sum_{i=1}^{n} w_{ij}\, X_i + \theta_j^h \quad (10)$$

$$X_i = (X_1, X_2, X_3, \dots, X_n)^t \quad (11)$$

Such as $i = 1,2, \dots, N_h$

x_i is the input vector applied to the layer, and w_{ij} is the weights of (i) input neuron connection, and θ_j^h represent the bias of hidden layer neurons.

The neurons of the hidden layer can be written as follows:

$$y_j^h = f(\sum_{i=1}^{n} w_{ij}\, X_i + \theta_j^h) \quad (12)$$

The output as:

$$y_k^0 = f(\sum_{i=1}^{n} w_{jk}\, y_j^h + \theta_k^0) \quad (13)$$

With θ_k^0: bias of neurons output layer.

Dynamic neural networks are good at time-series prediction. For instance, the NARX networks can be developed in open-loop form, closed-loop form, and open/closed-loop form. Using the Neural Network Time Series App in MATLAB. The first step, open the Neural Network Start GUI with this command: start. There are many selections such as Input-output and curve fitting, pattern recognition and classification, clustering, and dynamic time series as shown in Fig. 10.

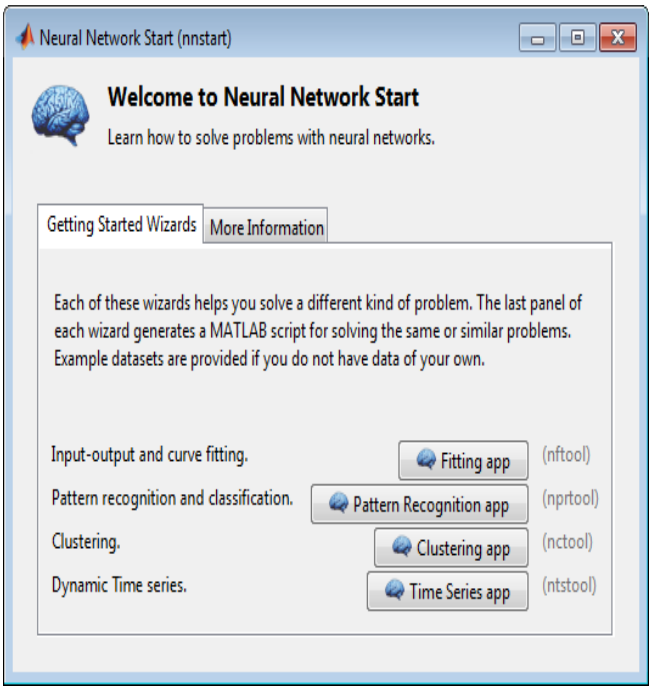

Fig. 10. Neural Network start GUI.

The second step, select Time Series App to open the Neural Network Time Series App. The NARX model will provide better predictions than this input-output model, because it uses the additional information contained in the previous values of $y(t)$ so, the first type will be chosen as displayed in Fig. 11. This form of prediction is called nonlinear autoregressive with exogenous (external) input, or NARX.

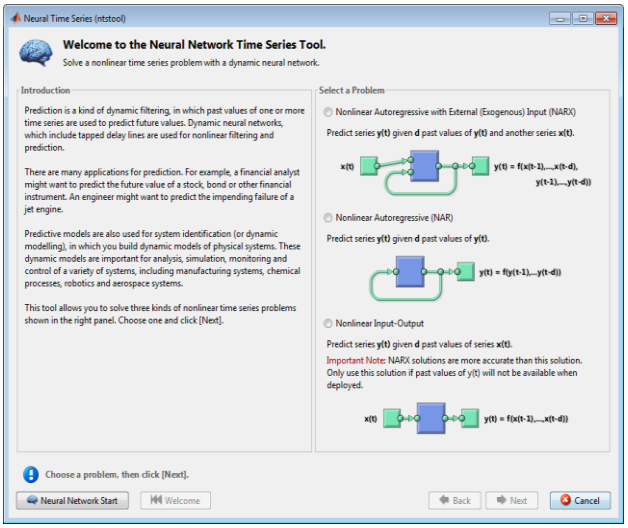

Fig. 11. Neural Network time series GUI.

The toolbox then provides a dialog box to define the network architecture. By default, the toolbox proceeds with a single hidden layer with a sigmoid activation function and a linear activation function for the output layer as shown in Fig. 12. This network also uses tapped delay lines to store previous values of the x(t) and y(t) sequences. Note that the output of the NARX network, y(t), is fed back to the input of the network (through delays), since y(t) is a function of y(t – 1), y(t – 2), ..., y(t – d). For this study, a hidden layer with 10 neurons was decided based on a literature study and multiple iterations.

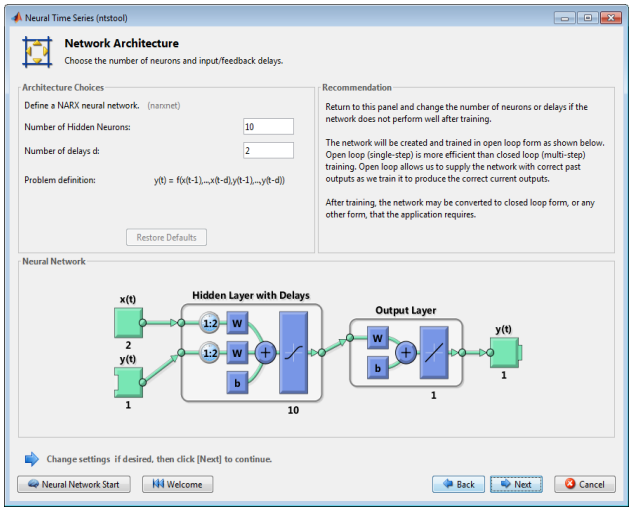

Fig. 12. Neural Network time series Architecture

Select a training algorithm from the following window as shown in Fig. 13, then click Train. Levenberg-Marquardt (trainlm) is recommended for most problems, Levenberg-Marquardt Algorithm that trains a neural network 10 to 100 times faster than the usual gradient descent backpropagation method. It always calculates the approximate Hessian matrix, which has dimensions n-by-n.

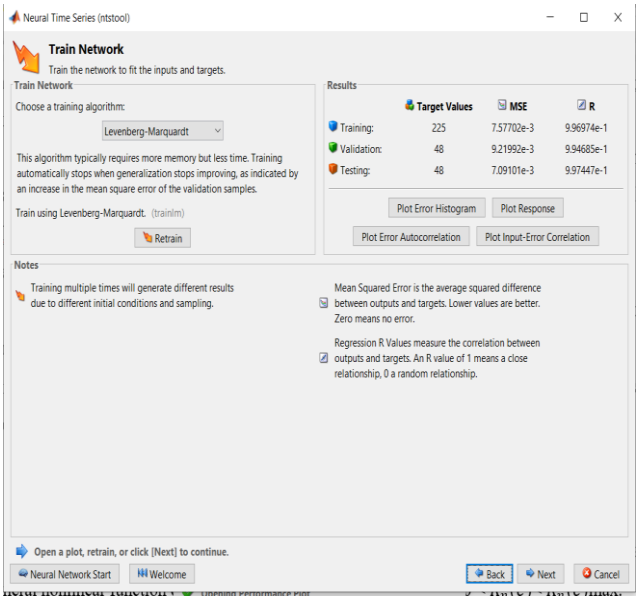

Fig. 13. Neural Network time series training algorithm.

The toolbox by default divides the data randomly into three categories: Training (70%), Validation (15%), and Testing (15%). The training data, as the name suggests, is used to train the network through an iterative process while the weights are adjusted during backpropagation based on the error calculated after each epoch. The validation data is used to determine if the network has achieved the required generalization and decides if the training can be stopped and overfitting is avoided. The testing data is used as an independent measure to evaluate the prediction performance of the trained neural network. It provides a measure of the mean squared error and R-value to assess how well the network has trained. The network is now ready to be trained.

Several procedures exist for estimating the network parameters. In neural network literature, the algorithms are called learning or teaching algorithms, in system identification, they belong to parameter estimation algorithms. The most well-known is back-propagation and Levenberg-Marquardt algorithms. Backpropagation is a gradient-based algorithm, which has many variants. Levenberg-Marquardt is usually more efficient but needs more capable computer memory. This algorithm does not require the calculation of the Hessien, it only approximates it.

$$H = J^T.J \tag{14}$$

The jacobian is calculated by

$$g = J^T.e \tag{15}$$

Where J is the jacobien matrix, which contains the first derivatives of the network errors with respect to the weights and biases, and (e) is a vector of network errors is given by:

$$e = \frac{1}{2}\sum_{i=1}^{n}(d_i^o - y_i^o)^2 \tag{16}$$

Where d_k^o is the desired output.

The jacobian matrix can be computed through a standard backpropagation technique that is much less complex than computing the hessian matrix. The adjust weights as:

$$\varphi_{k+1} = \varphi_k - [J^T.J + \mu I]^{-1} J^T.e \quad (17)$$

Where $\varphi = (w, \theta)$ and $I: Identity\ Matix.$

When the scalar μ is zero, this is just Newton's method, using an approximate Hessian matrix, when μ is large, this becomes gradient descent with a small step size. Newton's method is faster and more accurate near an error minimum, so the aim is to shift towards Newton's method as quickly as possible. Thus, μ is decreased after each successful step (reduction in performance function) and is increased only when a tentative step would increase the performance function. In this way, the performance function will always be reduced at each iteration of the algorithm.

If $\frac{e(k)}{e(k-1)} > 1 \Rightarrow \mu = \mu \times (\mu - inc)$ (18)

If $\frac{e(k)}{e(k-1)} < 1 \Rightarrow \mu = \mu \times (\mu - dec)$ (19)

Algorithm of Levenberg-Marquadt:

Stage 1: to initialize the weights and skews by small value random as well as the Parameters: m = 0,001(default value), m - dec= 0,1 (default value), m - Inc = 10 (default value),
Stage 2: To present the vector of input and desired output.
Stage 3: To calculate the output of the network by using the expression (12) & (13).
Stage 4: To calculate the error of output (16)
Stage 5: To calculate the error in the hidden layers
Stage 6: To calculate the gradient of the error compared to the weights
Stage 7: To calculate Hessien approximated by using the expression (15).
Stage 8: To adjust the weights according to the expression (17).
Stage 9: One test: by using the expression (18) & (19).
Stage 10: If the condition the error or the iteration count is reached or m reached m-max, outward journey at stage 11 if not outward journey at stage 2.
Stage 11: End

Fig. 14 demonstrates the linear speed of the actual experimental setup and the candidate models. It is noted that identified model based on DNN can simulate the behavior of the actual experimental setup compared to other identified models.

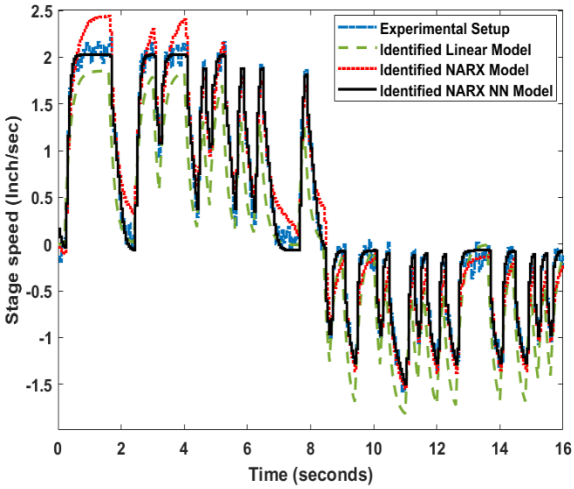

Fig. 14. The linear speed of one stage table servomechanism for actual experimental setup and identified models.

Table 2 demonstrates the mean square error of candidate models. It can be noted that identified model based on NARX neural network model has the minimum error compared to other identified models.

Table 2. Mean square error of each identified model.

Model type	Estimation to fit data	MSE
Identified linear model	79.624%	**0.0499**
Identified NAR Model	91.5 %	**0.00096**
Identified NAR neural network model	93.3%	**0.00071**

4. Conclusion

This paper shows the practical design steps of an identified model for one stage servomechanism drive system. The influence of nonlinear dead-zone caused by friction, backlash, and external disturbance is an obstacle to developing a mathematical model. So, we resort to constructing a nonlinear identified model by collecting the experimental input/output data and entering it into the MATLAB system identification toolbox. The obtained identified models were validated by comparing them with the actual input /output data. Also, a comparison between the candidate-identified models was implemented to investigate the validity of each model. The experimental and validation results provide that the identified model based on NARX deep neural network has the best performance with respect to other identified models.

References

[1] E. Yuliza, H. Habil, R. A. Salam, M. M. Munir, and M. Abdullah, "Development of a Simple Single-Axis Motion Table System for Testing Tilt Sensors," Procedia Eng., vol. 170, pp. 378–383, 2017.
[2] W. Lee, C. Lee, Y. Hun, and B. Min, "International Journal of Machine Tools & Manufacture Friction compensation controller for load varying machine tool feed drive," Int. J. Mach. Tools Manuf., vol. 96, pp. 47–54, 2015.
[3] P. Perz, I. Malujda, D. Wilczy, and P. Tarkowski, "Methods of controlling a hybrid positioning system using LabVIEW," Procedia Eng., vol. 177, pp. 339–346, 2017.
[4] P. Zhao, J. Huang, and Y. Shi, "Nonlinear dynamics of the milling head drive mechanism in five-axis CNC machine tools," Int. J. Adv. Manuf. Technol., 2017.
[5] C. Abeykoon, "Control Engineering Practice Single screw extrusion control : A comprehensive review and directions for improvements," Control Eng. Pract., vol. 51, pp. 69–80, 2016.
[6] M. Omar, M. A. Ebrahim, A. M, and F. Bendary, "Tuning of PID Controller for Load Frequency Control Problem via Harmony Search Algorithm," Indones. J. Electr. Eng. Comput. Sci., vol. 1, no. 2, pp. 255–263, 2016.
[7] M. A. Shamseldin, A. A. El-samahy, and A. M. A. Ghany, "Different Techniques of Self-Tuning FOPID Control for Brushless DC Motor," in MEPCON, 2016.
[8] M. A. A. Ghany, M. A. Shamseldin, and A. M. A. Ghany, "A Novel Fuzzy Self Tuning Technique of Single Neuron PID Controller for Brushless DC Motor," Int. J. Power Electron. Drive Syst., vol. 8, no. 4, pp. 1705–1713, 2017.
[9] F. Wang, C. Liang, Z. Ma, X. Zhao, Y. Tian, and D. Zhang, "Dynamic analysis of an XY positioning table," in International Conference on Manipulation, Manufacturing and Measurement on the Nanoscale (3M-NANO), 2013, no. 51205279, pp. 211–214.
[10] B. Feng, D. Zhang, J. Yang, and S. Guo, "A Novel Time-Varying Friction Compensation Method for Servomechanism," Hindawi Publ. Corp. Math. Probl. Eng., vol. 2015, p. 16, 2015.
[11] M. R. Stankovi, M. B. Naumović, S. M. Manojlović, and S. T. Mitrović, "FUZZY MODEL REFERENCE ADAPTIVE CONTROL OF VELOCITY SERVO SYSTEM," FACTA Univ. Ser. Electron. Energ., vol. 27, no. December, pp. 601–611, 2014.
[12] F. L. Li, M. L. Mi, and Y. Z. N. Jin, "Friction identification and compensation design for precision positioning," Springer, pp. 120–129, 2017.
[13] D. V. L. N. Sastry and M. S. R. Naidu, "An Implementation of Different Non Linear PID Controllers on a Single Tank level Control using Matlab," Int. J. Comput. Appl., vol. 54, no. 1, pp. 6–8, 2012.
[14] Y. Ren, Z. Li, and F. Zhang, "A New Nonlinear PID Controller and its Parameter Design," Int. J. Comput. Electr. Autom. Control Inf. Eng., vol. 4, no. 12, pp. 1950–1955, 2010.
[15] M. A. Ebrahim and F. Bendary, "Reduced Size Harmony Search Algorithm for Optimization," pp. 1–8.
[16] P. Zhao and Y. Shi, "Robust control of the A-axis with friction variation and parameters uncertainty in five-axis CNC machine tools," J. Mech. Eng. Sci., 2014.
[17] M. A. Shamseldin, M. A. Eissa, and A. A. El-samahy, "Practical Implementation of GA-Based PID Controller for Brushless DC Motor," in 17th International Middle East Power System Conference (MEPCON'15) Mansoura University,Egypt, December 15-17 ,2015, 2015.

BIBLIOGRAPHY OF AUTHORS

Dr. Mohamed.A. Shamseldin born in Cairo, Egypt, on October 1, 1987. He received the B.Sc. degree in mechatronics engineering in 2010 from faculty of engineering at Helwan, Helwan University, Cairo, Egypt. On December 2012, he received his work in faculty of engineering and technology at Future University in Egypt as an instructor in Mechatronics Engineering Department. He obtained M.Sc. in system automation and engineering management (2012 to 2016) from, Helwan University, Cairo, Egypt. In 2020, he obtained the Ph.D. in Mechatronics Engineering from faculty of engineering, Helwan University, Cairo, Egypt. Mohamed was visiting staff at University of Central Lancashire, Preston, UK. His research activity includes studying Artificial Intelligent techniques, electrical machines speed control and robot control.