

Design and Modelling of Ballscrew Linear Guide Actuator for Earthquake Shaking Table (EST) Based on Neural Network

Halimatus Sa'diyah¹, Purwadi Agus Darwito¹, Tavio², Murry Raditya³,
Gunawan Nugroho¹, and Fachrul Rizal²

¹Department of Engineering Physics, Institut Teknologi Sepuluh Nopember, Surabaya

²Department of Civil Engineering, Institut Teknologi Sepuluh Nopember, Surabaya

³Department of Instrumentation Engineering, Institut Teknologi Sepuluh Nopember, Surabaya
e-mail: halsdiy@gmail.com

Abstract—Earthquake Shaking Table (EST) is a device which can simulate an earthquake motion. This device is used to test the strength of a building structure against an earthquake motions before it's actually made. EST uses a variety of actuators one of them is ball-screw linear guide actuator. The EST used in this project is a bi-axial type which uses 2 linear actuators to simulate the x-axis and y-axis movement of earthquake, each of them used bipolar stepper motor as the main rotary-actuating device. This project models the linear guide actuator using backpropagation neural-network algorithm. The model is built with empirical method using datas taken from the real behavior of both linear actuators. The datas include acceleration, displacement, and velocity of both actuators and they are used to train the neural network using backpropagation with Levenberg-Marquadt method. Simulation is done using Simulink and the results show that model is able to produces nearly same exact movement with the real hardware with error approximately 0,214 % and 0,685% respectively for both actuators.

Keywords—Ballscrew Linear Guide Actuator, Earthquake Shaking Table, Neural Network.

I. INTRODUCTION

EARTHQUAKE shaking table abbreviated by EST, is an experiment platform which simulate earthquake motion and it also a device to test the strength of structure against an artificial earthquake. This artificial earthquake is produced by the table based on the data from the real earthquake. But not only a building structure or platforms that has been tested by this device, a railway tracks are also tested to determine the strength against the earth-vibrating motion [1-2]. In this study, a bi-axial EST that can produce two movements will be designed.

In designing a system, simulation is done first to avoid losses that occur if the system to be designed does not work well. In conducting simulations, the first thing needed is a system model. The system model is a representation of the actual system. The system model is obtained through the system modeling process. There are many types of system modeling, including mathematical modeling of the system. Mathematical modeling of the system produces mathematical models. The mathematical model is an abstraction of a system that is explained using mathematical branch languages including, for example; algebra, statistics, logic, and algorithms. This mathematical model was developed to function both as an experimental tool that can be used to

expand understanding of a system. in addition, this model can also be used as a prediction tool to assist in tasks such as decision making and automatic system control [3].

Mathematical models can be classified in many ways, including empirical models and theoretically derived models. The theoretical model is a model developed from what is considered a law or basic principle that governs system response. To get a theoretical model, it is necessary to have extensive knowledge of the law and basic principles of the system. As for the empirical model, the modeling requires a lot of experimental data to build the model. Empirical models are models developed from observing system responses that are being investigated for various situations. Often empirically modeling the system is referred to as a black box device. It is often assumed that empirical modeling provides little explanation of output results and there is no ability to extrapolate, and is therefore best used as a compromise in situations where a theoretical modeling framework is not available. But this characterization is somewhat unfounded. Empirical modeling is very important for developing theoretical models, making approaches more prevalent than might be realized. If used carefully, empirical modeling can provide insight into the internal structure and principles that drive the system. Modularization of the model can expand the scope of its application [3].

Empirical modeling has been widely used for system modeling. Empirical models of observational data using artificial neural networks have been proposed [4]. In his paper, Yerramareddy et.al, has shown how artificial neural networks have given them new tools for empirical modeling of observational data. They show that in terms of predictive accuracy, artificial neural networks models are better compared to regression models that are conventionally used for empirical models [4]. An empirical model of artificial neural networks for estimating phytoplankton production has been proposed [5]. In his paper, two conventional empirical models are compared with a new approach, based on artificial neural networks. Although the neural networks used are very simple, they provide a better fit for the observed data than conventional models. According to him, neural networks seem very promising for modeling phytoplankton production [5]. Empirical model and Artificial NN approach for Air Dried Sheets (ADS) rubber has proposed [6]. In his research, Ninchuewong et.al, compared the prediction results between

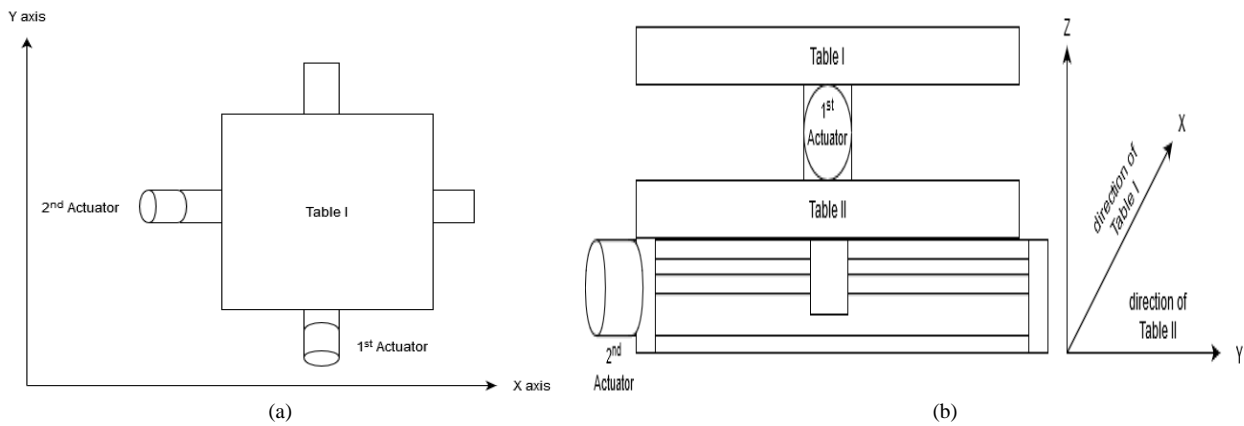


Figure 1. (a) Skema EST bi-axial tampak atas; (b) skema EST bi-axial tampak samping.

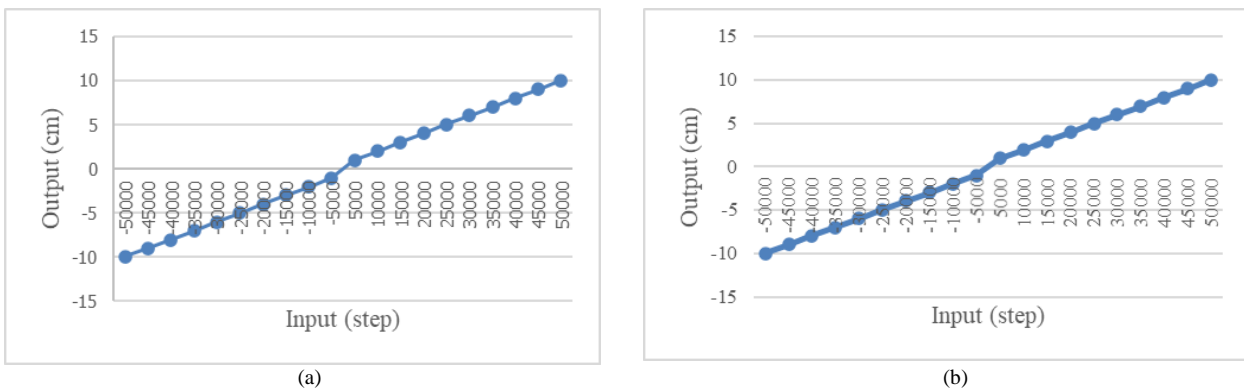


Figure 2. (a) Actuator 1 position data with 50000 step / s² acceleration input, 50000 step / s input speed and varying position in steps; (b) Actuator 2 position data with input acceleration of 50000 step / s² and input speed of 50000 step / s and position in varying steps.

empirical models using the Verma model with ANN. It was found that ANN can describe the drying behavior effectively. The empirical model approach for the truck weigh-in motion problem has been proposed by [7]. Wang et al studied the existing empirical modeling approach to solve the WIM Weigh-in-motion problem, specifically for the classification of truck types. The use of (support vector machines) SVM as a potentially more accurate alternative approach is proposed. The performance of six ANN and SVM-based truck classifiers was compared using truck weighing data generated synthetically. The optimal version of each model is determined using an empirical modeling parameter selection scheme based on LOESS. The results show that the SVM model significantly outperforms the ANN model in terms of the correct number of truck classifications [7]. In 2019, Li et.al proposed an empirical comparison of multiple linear regression and artificial neural networks for concrete and deformation modeling [8]. In their paper, they investigate the usefulness of the multiple linear regression (MLR) and ANN models in predicting dam deformation. The results of his study indicate that ANN produces higher prediction accuracy.

In this study, the bi-axial EST that will be designed uses ballscrew linear guide as its driving force. Bi-axial EST is driven by these two actuators. Each actuator moves EST in a different direction. The first actuator, moves the table in the direction of the X axis and the second actuator moves the table in the direction of the Y axis. Before designing a control system to control these two actuators so they can move together, a model is needed for simulation. The ballscrew

system is a non-linear system. To model a ballscrew system, sufficient knowledge is needed to determine physical parameters, for example motor inertia, shaft inertia, stiffness coefficient, friction coefficient, damping coefficient, viscous damping etc. These parameters are not included in the datasheet. Need to identify the system to get it. With empirical modeling, modeling the system without knowing these parameters is very possible. With experimental data, empirical models can be obtained. In addition, with empirical modeling, hardware characteristics can be modeled in full based on experimental data.

In this study, modeling of ballscrew linear guide actuators to drive bi-axial ESTs was carried out. Modeling this actuator system using Artificial Neural Network. Based on some literature, NN is a pretty good black box tool for system modeling. NN is very good for pattern recognition. Actuator modeling using NN begins with data input and output retrieval. Some of this data is used for training, and others for validation. NN used for modeling is backpropagation with the Lavenberg-Marquadt method. To get the best model, four NN model frameworks were created, each of which was NN with 2, 5, 10 and 50 hidden neurons. Of the four NN models, the one with the smallest error was chosen as the actuator model.

II. METHOD

The purpose of this study is to model the actuator system, namely ballscrew linear guide actuator, using NN. Previously, to clarify the system that would be designed, in

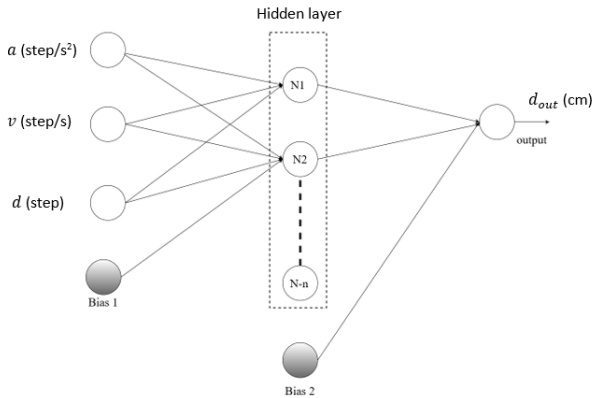


Figure 3. Schema of artificial neural networks.

Figure 1. The EST bi-axial system scheme to be designed was shown. Figure 1 shows that the bi-axial EST was designed, consisting of two tables and two actuators.

Figure 1 shows the scheme of the two-axis EST designed in this study. The two-axis EST consists of 2 tables and 2 movers. Table I is moved by actuator 1 in the direction of the X axis, while table II is moved by actuator 2 in the direction of the Y axis. To produce motion that represents earthquake motion to the two axes (X and Y), the two axis EST must be able to be moved simultaneously in the direction the axis For example, at the same time t, Table 1 must move 2 cm in the direction of the X axis and Table 2 must move 3 cm in the direction of the Y axis. So when Table 2 moves as far as 3 cm in the direction of the Y axis, drive 1 and Table 1 becomes the load of the movers 2. So that the movers 1 and Table 1 come to move as far as 3 cm to the Y axis with the condition that Table 1 is also being moved by movers 1 move as far as 2 cm to the X axis.

The actuators used to drive this EST have been determined, namely using the FLS80 type ballscrew linear guide actuator produced by FUYU. This drive consists of a stepper motor and ballscrew. Modeling a ballscrew linear guide actuator system with a table into a mathematical model is quite difficult to do. Some parameters such as motor inertia, viscous damping coefficient, stiffness coefficient, are not included in the datasheet. Therefore, in this study, empirically modeling the system was chosen to model this system.

Empirical system modeling is done by approaching the investigation and taking the position data from each table. Investigations are carried out on slips that can occur at any time after the actuator is given a certain position input. Data is collected by recording the position input entered into the system and the resulting position output. After taking input and output data from actuator 1 and actuator 2, the ANN model framework is created, then continued with training data. Model ANN actuator 1 and actuator 2 which produce the smallest error selected. After that the model is tested with the same input as the one entered into the real system. Model results and system real are compared. If the model output error with real system output is <5%, then the Table 1 and Table 2 models using ANN can be used to model the two-axis EST system.

To move the ballscrew linear guide actuator, stepping drivers are needed. The stepping driver used on this EST is FUYU FMDD50D40NOM. From the FUYU

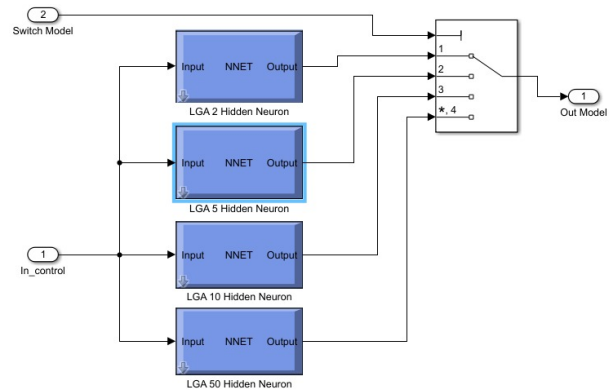


Figure 4. Actuator models using ANN with a variation of 4 hidden neurons.

FMDD50D40NOM datasheet it is known that the driver has 5000 step/revolution capabilities. 5000 step/revolution in question is to produce 1 round of 5000 step motors are needed. To produce 1 step motor, it requires 1 signal wave with a duty cycle of approximately 5µs.

Lead is the linear distance produced by a screw or nut for one full rotation. FLS80 has 10mm leads, meaning that one full rotation of the linear mileage produced by the screw is 10mm or 1 cm. While the stepping driver has the ability to produce 1 full rotation it takes 5000 steps. So, to produce 1 full rotation the controller must signal as many as 5000 signals. Thus, it was found that to move the actuator as far as 1 cm required 5000 steps.

$$1\text{cm} = 5000\text{ step} \tag{1}$$

In operation, the actuator requires an input step. So, it needs to be converted from cm to step first. Equation (1) is the conversion equation from cm to step. In data retrieval, the position, speed and acceleration that will be used as input to the system are converted into steps. Then the position produced by the actuator is recorded.

The data collection of the two actuators is carried out 5 times in each variation of the acceleration, speed and position input. The variation of acceleration input given is 50000 steps/s², 80000 steps/s² and 100000 steps/s². The input speed variations provided are 50000 step/s, 80000 step/s and 100000 step/s. While the given position variations are -50000 steps, -45000 steps, -40000 steps, -35000 steps, -30000 steps, -25000 steps, -20000 steps, -15000 steps, -10000 steps, -5000 steps, 5000 steps, 10000 step, 15000 step, 20000 step, 25000 step, 30000 step, 35000 step, 40000 step, 45000 step and 50000 steps. Figure 2 is a snapshot of actuator 1 and actuator 2 data.

Figure 2 is a part of the input and output data that will be used as training data on system modeling using ANN. To get the EST model using ANN, what needs to be done first is to model each table system consisting of a table and actuator, based on the direction of motion. This EST consists of 2 systems, namely actuator 1 and actuator 2. The EST model is obtained from a combination of actuator 1 and actuator 2 models.

A. Modeling of 1st Actuator using NN

Modeling actuators 1 using ANN will be discussed in this section. After input and output data are obtained, the ANN

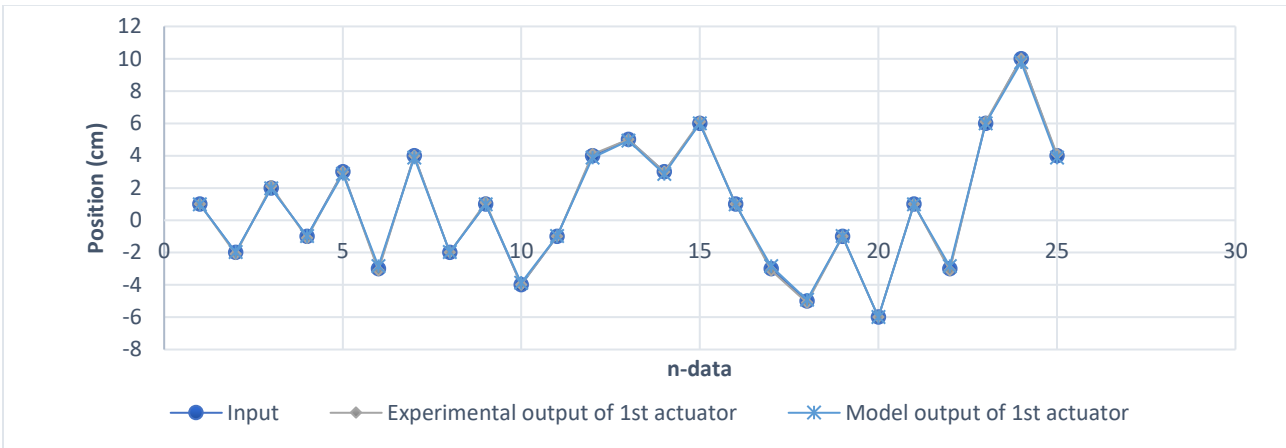


Figure 5. Result of 1st actuator test with randomized input.

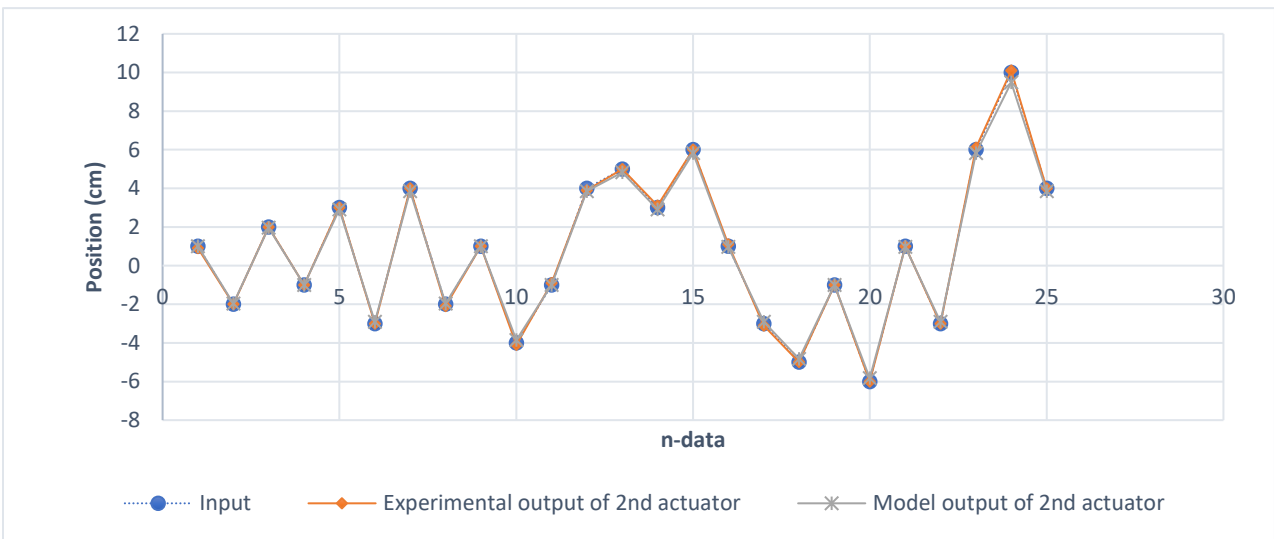


Figure 6. Results of 2nd actuator test using randomized input.

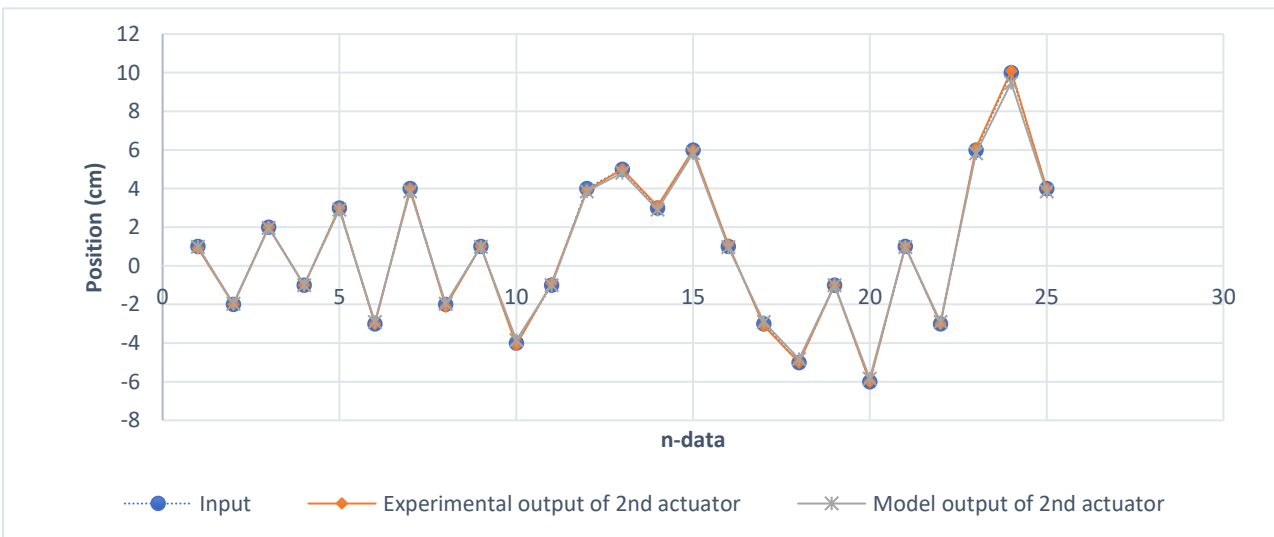


Figure 7. Results of 2nd actuator test using randomized input

model framework with 1 hidden layer is designed as shown in Figure 3.

In this ANN modeling, 4 types of models with different number of hidden neurons are designed, namely: 2, 5, 10 and 50 neurons in a row and later, from the four models one model will be selected which produces the smallest error. The model designed is a multi-layer perceptron multi-layer ANN model

with back propagation. The ANN modeling steps are as follows.

First, initial initialization of all weights with random values. Then the feed forward process is carried out using the equation:

$$d_{out} = f^i(x) \tag{2}$$

Table 1.
Test results for 1st actuator model.

1 st Actuator Model	Input	Output	% Error
2 Hidden Neuron	9	8,242	8,426
5 Hidden Neuron	9	9,019	0,214
10 Hidden Neuron	9	9,056	0,624
50 Hidden Neuron	9	9,056	0,624

Table 2.
Result of 2nd table test.

2 nd Table Model	Target	Output	% Error
2 Hidden Neuron	9	7,979	11,35
5 Hidden Neuron	9	9,062	0,685
10 Hidden Neuron	9	8,852	1,639
50 Hidden Neuron	9	9,117	1,302

with,

$$x = \sum w^i x_i \tag{3}$$

$$f^i = \frac{1 - e^{-2x}}{1 + e^{-2x}} \tag{4}$$

Where w^i is weights for each layer, x_i is the input value for each layer, and f^i is sigmoid bipolar activation function. From the output flow forward process, an error is obtained against the target, e . By using the error obtained, back propagation is done to update the original weight using the equation as follows.

$$w_{(k+1)} = w_{(k)} + \alpha_{(k)} e_{(k)} x_{(layer,i)} \tag{5}$$

Where $w_{(k+1)}$ is new weights, $\alpha_{(k)}$ is learning rate, $e_{(k)}$ is error dan $x_{(layer,i)}$ is input for each layer. Then the flow is carried forward as before, then repeating until the smallest error is obtained.

Actuator model 1 is a system model of Table 1 which is driven by drive 1 in the direction of the X axis which modeling using ANN. The design of the actuator 1 model using software is shown in Figure 4.

Actuator 1 system is modeled using ANN. The design of actuator model 1 is shown in Figure 4. Figure 4 is an ANN model that has been designed with 4 hidden neuron variations, namely 2, 5, 10 and 50 hidden neurons. After the actuator 1 model is designed, this model is tested by entering the input values of step/s², step / s, and step. The output value generated by the model is compared with the real system output. Then calculated the% error for each hidden neuron variation. The best model is the model that produces the smallest error value. The test results of actuator 1 model design are shown in Table 1.

Table 1 is the result of testing the actuator model 1 using ANN by entering the input values of 50000 steps/s², 50000 steps/s and 45000 steps. The smallest percentage error value is 0,214%, obtained by a model with 5 hidden neurons. After obtaining the model with the smallest error, a decrease in the equation of the ANN model with 5 hidden neurons has been obtained. The steps taken to derive the equation are as follows.

First, retrieve data that has been updated for each layer. Using equation (2), the equation for each neuron in the hidden layer is obtained as follows:

Table 3.
The results of the comparison of actuator 1 model output and real actuator 1 system.

Input cm	Real Hardware Output of 1 st Actuator System (cm)	Model Output of 1 st Actuator(cm)	Error Between Real Hardware's Output and Models (%)
	1	1,04	0,989
2	2,02	1,97	2,475
3	3	2,853	4,9
4	4,02	3,876	3,582
5	5,04	4,931	2,216
6	6,04	5,999	0,828
7	7,04	7,05	0,142
8	8,04	8,049	0,112
9	9,02	8,966	0,598
10	10,02	9,777	2,425

Table 4.
The results of the comparison of the output of the actuator 2 model and the real actuator 2 system.

Input cm	Output Sistem Real Aktuator 2 (cm)	Output Model Aktuator 2 (cm)	Error Antara Output Sistem Real dengan Model Aktuator 2 (%)
1	1	0,997	0,3
2	2	1,961	1,95
3	3	2,908	3,067
4	3,9	3,855	1,154
5	5	4,824	3,52
6	5,9	5,831	1,169
7	7	6,872	1,829
8	8	7,898	1,275
9	8,9	8,807	1,045
10	9,9	9,482	4,222

$$Neuron1 = -4,12337a + 1,362553v - 0,82458x + 2,562103 \tag{6}$$

$$Neuron2 = -0,90542a + 0,099421v - 1,42895x - 1,73994 \tag{7}$$

$$Neuron3 = 0,686861a - 0,1464v - 0,21421x - 0,36854 \tag{8}$$

$$Neuron4 = -0,5435a + 2,936556v - 0,67484x - 1,32072 \tag{9}$$

$$Neuron5 = -0,54937a + 0,084257v - 1,52868x + 0,40696 \tag{10}$$

Where a is the acceleration value input, v is the speed value input, x is the position value input. By substituting each equation (6) to equation (10) into the activation function, equation (3), a calculation is made for each neuron, and the following equation is obtained:

$$N_{(1)} = \frac{1 - e^{-2(-4,12337a + 1,362553v - 0,82458x + 2,562103)}}{1 + e^{-2(-4,12337a + 1,362553v - 0,82458x + 2,562103)}} \tag{11}$$

$$N_{(2)} = \frac{1 - e^{-2(-0,90542a + 0,099421v - 1,42895x - 1,73994)}}{1 + e^{-2(-0,90542a + 0,099421v - 1,42895x - 1,73994)}} \tag{12}$$

$$N_{(3)} = \frac{1 - e^{-2(0,686861a - 0,1464v - 0,21421x - 0,36854)}}{1 + e^{-2(0,686861a - 0,1464v - 0,21421x - 0,36854)}} \tag{13}$$

$$N_{(4)} = \frac{1 - e^{-2(-0,5435a + 2,936556v - 0,67484x - 1,32072)}}{1 + e^{-2(-0,5435a + 2,936556v - 0,67484x - 1,32072)}} \tag{14}$$

$$N_{(5)} = \frac{1 - e^{-2(-0,54937a+0,084257v-1,52868x+0,40696)}}{1 + e^{-2(-0,4937a+0,084257v-1,52868x+0,40696)}} \quad (15)$$

Then the output layer calculation is done with equation (2) as follows:

$$x_{out_meja1} = f(-0,10706N_{(1)} - 0,61244N_{(2)} - 0,77742N_{(3)} + 0,014044N_{(4)} - 0,58826N_{(5)} - 0,48048) \quad (16)$$

So, the output equation of table model 1 with 5 hidden neurons is as follows:

$$x_{out_meja1} = \frac{1 - e^{-2(-0,10706N_{(1)} - 0,61244N_{(2)} - 0,77742N_{(3)} + 0,014044N_{(4)} - 0,58826N_{(5)} - 0,48048)}}{1 + e^{-2(-0,10706N_{(1)} - 0,61244N_{(2)} - 0,77742N_{(3)} + 0,014044N_{(4)} - 0,58826N_{(5)} - 0,48048)}} \quad (17)$$

From equation (17) which has been obtained for the actuator 1 model, a block model is designed using Simulink.

B. 2nd Actuator Modelling using NN

Actuator 2 model is a system model of Table 2 which is driven by drive 2 in the Y axis direction which is modeling using ANN. The results of actuator 2 model design use the same software as shown in Figure 4. Both actuator 1 and actuator 2 models, the design of the ANN model is the same, namely with 4 variations of hidden neurons, namely 2, 5, 10 and 50 hidden neurons. Like testing the ANN model of actuator 1, from the model that has been designed, then tested by entering the value of the input acceleration in step/s², speed in step / s, and position in step. The model output is compared with the real actuator 2 system output, so we get % error. The model chosen is the model that produces the smallest error value. The results of testing the design of actuator model 2 are shown in Table 2.

Table 2 is the result of testing the actuator 2 model by entering the input values of 50000 steps/s², 50000 steps/s and 45000 steps. It was found that the value of the smallest % error was obtained from actuator model 2 which used 5 hidden neurons with a value of 0,685%. After obtaining the model with the smallest error, a decrease in the equation of the ANN model with 5 hidden neurons has been obtained. The steps in the modeling Table 1 using ANN are carried out for Table 2 so that the following equation is obtained:

$$N_{2(1)} = \frac{1 - e^{-2(1,61944a-0,07122v-1,58156y-1,96977)}}{1 + e^{-2(1,61944a-0,07122v-1,58156y-1,96977)}} \quad (18)$$

$$N_{2(2)} = \frac{1 - e^{-2(4,462311a-0,00418v+0,513271y-0,27102)}}{1 + e^{-2(4,462311a-0,00418v+0,513271y-0,27102)}} \quad (19)$$

$$N_{2(3)} = \frac{1 - e^{-2(-0,31014a-0,06224v+2,037134y-3,31046)}}{1 + e^{-2(-0,31014a-0,06224v+2,037134y-3,31046)}} \quad (20)$$

$$N_{2(4)} = \frac{1 - e^{-2(1,400317a+0,001445v-0,71521y+0,917161)}}{1 + e^{-2(1,400317a+0,001445v-0,71521y+0,917161)}} \quad (21)$$

$$N_{2(5)} = \frac{1 - e^{-2(0,430551a+0,249412v+4,340352y+5,654888)}}{1 + e^{-2(0,430551a+0,249412v+4,340352y+5,654888)}} \quad (22)$$

By substituting equations (18) to (21) to equation (2), the equation of actuator 2 model is obtained as follows:

$$x_{out_meja2} = \frac{1 - e^{-2(-0,56043N_{2(1)} + 1,336893N_{2(2)} + 4,396079N_{2(3)} - 1,57732N_{2(4)} + 3,219724N_{2(5)} + 1,419813)}}{1 + e^{-2(-0,56043N_{2(1)} + 1,336893N_{2(2)} + 4,396079N_{2(3)} - 1,57732N_{2(4)} + 3,219724N_{2(5)} + 1,419813)}} \quad (23)$$

From equation (23) that has been obtained for Table 2 model, a block model is designed using Simulink.

III. RESULT AND DISCUSSION

After the actuator 1 and actuator 2 models are obtained, then the validation of actuator 1 and actuator 2 models is carried out. Validation is done by comparing the output produced by the model, both actuator 1 and actuator 2 models, with the output of real actuator 1 and actuator 2. Inputs entered into the model and real actuator system are the same. The model output and the actuator system real are recorded, then compared. The error value between the real system output and the actuator model is calculated. The results of actuator 1 ANN model testing are shown in Table 3. While the results of actuator 2 ANN model testing are shown in Table 4. Table 3 shows the results of comparison between positions produced by actuator model 1 and positions produced by real system actuator 1. From Table 3 can be seen that the biggest error was 4,904%, namely at 50000 steps/s², 80000 steps/s, 5000 steps.

Testing of actuator model 2 was also carried out for model validation. Validation is done by comparing the output of actuator model 2 with the output of the real actuator system 2. The input given and the output generated by the model are shown in Table 4.

Table 4 shows the results of a comparison between the positions generated by the actuator 2 model and the positions generated by the real actuator 2 system. From Table 4 it can be seen that the largest error is 4.222%, namely at the input 50000 step /s², 80000 steps/s 50000 steps.

The next test is to test the actuator 1 and actuator 2 models with random position data. The results of testing on actuator 1 are shown in Figure 5. While the results of testing with random position data input on actuator 2 are shown in Figure 7.

By entering 25 position data into actuator models 1 and 2, the results obtained as shown in Figure 6 and Figure 7. Based on these two figures, it is obtained that the actuator 1 and actuator 2 models have produced position values that are close to the real value of the actuator system. Judging from the errors that resulted in testing this actuator, it was found that both actuators produced errors of less than 5%. The average position position generated by actuator 1 is 4,025% and 4,225% for 2nd actuators.

Thus, actuator models 1 and 2 that will be used in the design of the Earthquake Shaking Table have been obtained through empirical modeling, namely using Neural Networks. However, because this model is an empirical model, this model cannot be used freely in other conditions. Neural Network models depend on training data. If this model is used for purposes with data outside of NN's knowledge, an error will occur in the output that will be generated by the NN model. In this study, EST model training data uses position input with round values, which are 1 cm, 2 cm, 3 cm and so on. In addition, position data trained are only up to 10cm. Thus, the model will not work well if it is used on input values greater than 10cm.

IV. CONCLUSION

Actuator modeling, namely ballscrew linear guide actuator has been carried out using empirical modeling. Neural Network is used to model actuator 1 and actuator 2 which will be used to drive bi-axial EST. The models for both actuators can optimally simulate the behavior of real ball screw linear guide actuators. They are designed using neural networks and are able to achieve precise movement with errors approximately 4,025% for 1st actuators and 4,225% for 2nd actuators.

REFERENCES

- [1] T. Baran, A. K. Tanrikulu, C. Dundar and A. Tanrikulu, "Construction and performance test of a low-cost shake table," *Experimental Technique*, pp. 8-16, 2011.
- [2] R. T. Severn, "The development of shaking tables-A historical note," *Earthquake Engineering and Structural Dynamics*, vol. 40, pp. 195-213, 2011.
- [3] I. Flood and R. R. A. Isaa, "Empirical Modeling Methodologies for Construction," *Journal of Construction Engineering and Management*, vol. 136, no. 1, pp. 36-48, 2010.
- [4] S. Yerramareddy, S. C.-Y. Lu and K. F. Arnold, "Developing Empirical Models from Observational Data Using Artificial Neural Networks," *Journal of Intelligent Manufacturing*, pp. 33-41, 1993.
- [5] M. Scardi, "Artificial neural networks as empirical models for estimating phytoplankton production," *Marine Ecology Progress Series*, vol. 139, pp. 289-299, 1996.
- [6] . T. Ninchuewong, S. Tirawanichakul and Y. Tirawanichakul, "Empirical Model and Artificial Neural Network Model Approach for Air Dried Sheets (ADS) Rubber," *Advanced Materials Research*, Vols. 622-623, pp. 69-74, 2013.
- [7] Y. Wang and I. Flood, "Empirically-Based Modelling Approaches To The Truck Weigh-In-Motion Problem" in *Proceedings of the 2015 Winter Simulation Conference*, 2015.
- [8] M. Li and J. Wang, "An Empirical Comparison of Multiple LinearRegression and Artificial Neural Network forConcrete Dam Deformation Modelling," *Mathematical Problem in Engineering*, vol. 2019, pp. 1-13, 2019.