

# The Artificial Power System Networks Stability Control Using the Technology of Neural Network

A M-N Alzakkar<sup>1,\*</sup>, I M Valeev<sup>1</sup>, N P Mestnikov<sup>2</sup> and E G Nurullin<sup>2</sup>

<sup>1</sup>Institute of Electrical Engineering, KSPEU, Kazan, Krasnoselskaya Street 51, Russia

<sup>2</sup>Institute of Heat Power Engineering, KSPEU, Kazan, Krasnoselskaya Street 51, Russia

**Abstract.** In the present work, the electric voltage stability at Muharda station in Syria was studied during the normal and up to normal loading states. The results were obtained using artificial neural network, which consists of three layers (input-hidden-output). This network is characterized by the speed and accuracy in processing before failure and supply turn-off, which may lead to economical problems. This study was carried out using two different generating schemes in this station (single - double). The performance of this network consists of two stages: training stage (off-line) and testing stage (on-line), and a comparison between these stages is carried out, which leads to optimization the load in testing cases depending on the training data.

## 1 Introduction

The stability of an electric power system is a term in electrical power engineering that represents the ability of the system to return to the normal state of operation (at the default voltage and frequency) after a disturbance [1-3]. Several methods are used for that purpose like equal area criterion [4] and transient stability margin (TSM) [5]. The protection of individual systems is relatively simple when the directional voltage protection is used.

Increased size of the networks and complicated connection of these networks drives the need to create an artificial neural network [6], which is used for analyzing the system feedback and processing in relatively short time in comparison with the other methods to avoid the voltage collapse [7,8] in the electrical system.

## 2 Construction of the neural network

The artificial neural network consists of three layers as shown in figure 1.

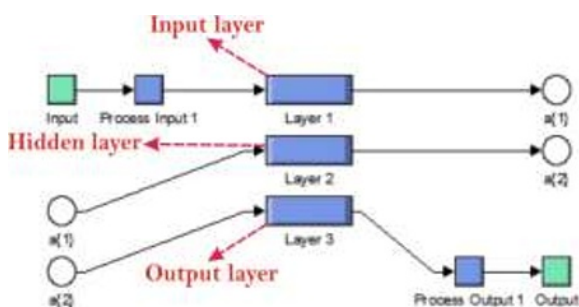


Fig. 1.

1. Input layer: This is an important part where the voltage stability assessment requires accurate data in the training stage to the input layer. According to that information, the critical situations in the electrical system can be processed.

2. Hidden layer: The number of neurons in the hidden layer ranges from 3 to 12 because the values are nonlinear.

3. Output layer: The output layer in this situation consists of a single neuron. The aim of this layer is to predict the value of the maximum load depending on the data obtained from the training stage [9,10].

The neural network implementation has the following steps:

1. The training stage (off-line)

In the present work, the neural network was trained using Back Propagation Algorithm (BPA) [11-13] as training algorithm, which is one of the best algorithms used for the statistically cases studying (off-line). This network has been applied to a test system for Muharda station in Syria, which has four turbines where each of them has the nominal capacity of approximately 150 MW. Two cases are studied in this work shown in table 1.

Figure 2 shows the program interface that was designed using the Matlab software [14,15]:

\* Corresponding author: [ahmadalzakkar86@gmail.com](mailto:ahmadalzakkar86@gmail.com)

**Table 1.**

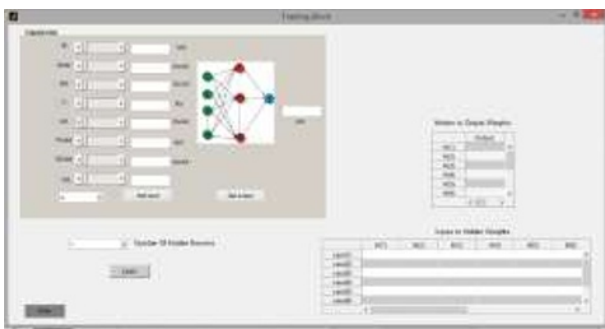
Input Vectors	Single generator (G <sub>1</sub> )	Double generator (G <sub>1</sub> +G <sub>2</sub> )
Generated power P <sub>G</sub> (MW)	110	225
Maximum reactive power Q <sub>max</sub> (Mvar)	25	47
Reactive power reserve R <sub>Q</sub> (Mvar)	13	20
Voltage V (kV)	230	230
Total demand from reactive power Q <sub>T</sub> (Mvar)	20	45
Real power losses P <sub>Loss</sub> (MW)	2	4
Reactive power losses Q <sub>Loss</sub> (Mvar)	1	3
P <sub>Load</sub> (MW)	100	190

**Table 2.**

Cases	Input to Hidden Weights											
	H(1)	H(2)	H(3)	H(4)	H(5)	H(6)	H(7)	H(8)	H(9)	H(10)	H(11)	H(12)
1	-0.37	0.257	0.028	-0.18	0.140	0.545	0.243	0.058	0.222	-0.05	-0.76	0.355
2	0.995	-0.29	0.942	-0.31	0.768	-0.09	-0.17	-0.56	-0.74	-0.38	0.456	0.560
3	0.361	-0.92	0.700	0.849	0.584	-0.94	-0.29	0.435	0.423	-0.50	-0.51	0.399
4	-0.13	0.473	-0.46	-0.62	0.799	0.435	0.700	-0.68	-0.76	-0.63	-0.19	-0.429
5	0.781	-0.77	-0.90	0.117	0.561	-0.39	-0.66	-0.31	-0.59	0.043	0.791	0.280
6	-0.86	-0.04	-0.84	0.018	-0.02	0.918	0.491	0.049	0.692	-0.59	-0.35	1.001
7	0.843	0.403	0.230	-0.31	0.857	-0.74	0.478	0.282	0.678	-0.22	0.517	0.651
8	-0.36	0.125	0.961	0.099	-0.32	0.231	-0.29	0.521	-0.18	-0.002	0.373	0.963

**Table 3.**

Result	Hidden to output weights											
	H(1)	H(2)	H(3)	H(4)	H(5)	H(6)	H(7)	H(8)	H(9)	H(10)	H(11)	H(12)
Output	0.205	-0.57	-0.82	-0.24	0.170	0.549	0.480	-0.06	-0.49	0.441	1.761	-0.714



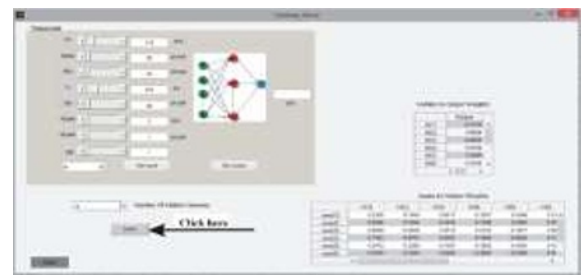
**Fig. 2.**

It's clear that figure 2 contains two tables; the first table represents the change rate of neural weights during the motion from the input layer to the hidden layer. The neural weights represent the unit on which the network depended in loading comparison process. The second table represents the rate of change of neural weights during the motion from the hidden layer to output layer.

After training of the neural network, these values are taken from the system and are inserted in the following tables.

By clicking on the learn icon, the neural network begins the training process according to the input data, as

shown in the following example (Figure 3), which shows the results after the neural network training:



**Fig. 3.**

**2. Testing Stage (on-line):**

Testing in case of a single generator (G<sub>1</sub>): in this case a single generator feeds the total load of the network; and the highest load in that case was 100 MW.

**A-Overload situation**

The values in the first table must be sent to the program when clicking the "Get output" icon. The output value in this situation is 100.0255, which is the output of the neural network and the new window with two options will extrude, the first shows the currently measured load and the second shows the effect of load on the network depending on the input data.

The output state can be understood as follows: if the immediately measured power is 150 MW which is greater than the greatest generated power, this situation will lead to the collapse of electrical voltage [6,7] and the sign “NO” appears in the red rectangle of conversation window “Black out” as shown in figure 4.

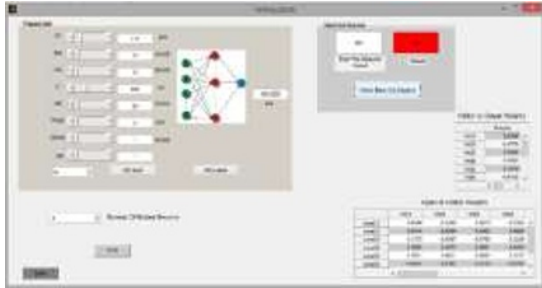


Fig. 4.

**B-Normal load situation**

Figure 5 shows the situation when the load is equal to or less than the peak power supply which was 100 MW in this case. For example, if the loading of that network is 85 MW, therefore the program will refer to turning on the situation by the sign “YES” in green rectangle representing the system stability. .

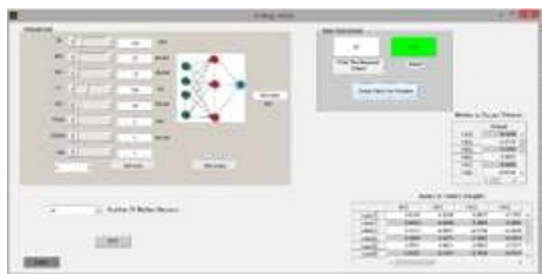


Fig. 5.

Testing in case of double generators (G1+G2):

When the first and the second generators work together to feed the total load of the network, the greatest load is achieved (190 MW).

**A-Overload situation**

The values in the first table must be sent to the program when clicking the “Get output” icon. The normal output from this network is 190.0176 MW. If the measured load exceeds the value of 250 MW, the electrical voltage will collapse and the “NO” sign in red rectangle appears in conversation window “Black out” as in figure 6.

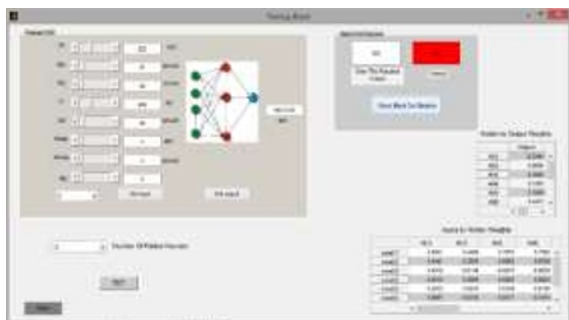


Fig. 6.

**B-Normal load situation**

Figure 7 shows the status when the load is equal to or less than the generated power (190 MW). Now, if loading is 175 MW, the system is normally supplying the network and the sign “YES” will appear in the green rectangle referring to the stability of the network.

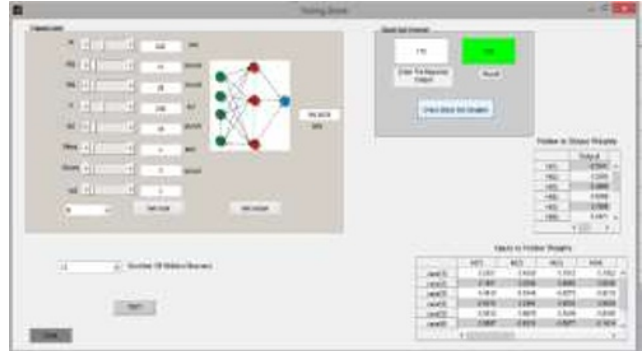


Fig. 7.

Figure 8 shows the controlling ability of the neural network on the output of the system depending on the accurate information about the system on which the network must be trained.

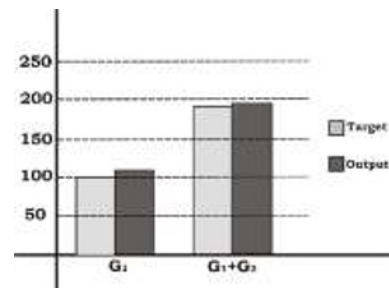


Fig. 8.

**3 Conclusion**

It was found that when training the neural network, a very close and accurate information about the studied system is needed. The response and processing time are rather shorter than that in case of using the other (traditional) methods for network stability controlling.

**References**

1. Prabha Kundur, *Power System Stability and Control*, Electric Power Research Institute, 1176 (1993)
2. Huabo Shi, Xiaoyan Zhou, *Stability Analysis on Power System with Large Power Source*, Scientific Research (2013)
3. Prabha Kundur, John Paserba, Venkat Ajarapu and Göran Andersson, *Definition and Classification of Power System Stability*, IEEE Transactions on power systems (2004)
4. Sumit Paudyal, Gokaraju Krishna and Mohindar S.Sachdev, *Application of Equal Area Criterion Conditions in the Time Domain for Out-of-Step Protection*, IEEE Trans (2010)

5. H.El-Khazraji, Assessment of Power System Transient Stability by (TSM) Method, Iraqi academic scientific journals (2011)
6. Crescenzo Gallo. Artificial Neural Network: tutorial, <https://www.researchgate.net>, 189 (2017)
7. M. Sachdev, H. Shuh, P. Solanics and J. Williams, Voltage Collapse Mitigation, Report to IEEE Power System Relaying Committee, 36 (1996)
8. Isaac Samuel, Aremu Cladius Ojo, James Katende and Ayokunle Awelewa, Prediction of Voltage Collapse in Electrical Power System Networks using a New Voltage Stability Index, International Journal of Applied Engineering Research (2017)
9. James A. Momoh and Mohamad E.El-Hawary, *Electric Systems Dynamics & Stability with Artificial Intelligence Applications*, Dalhousie University Halifax, 356 (2000)
10. Baba Shehu Waziri, Kabbir Bala and Shehu Bustani. Artificial Neural Networks in Construction Engineering and Management, International Journal of Architecture (2017)
11. Ajoy Kumar Palit and Gerhard Doeding. Backpropagation Based Training Algorithm for Takagi - Sugeno Type MIMO Neuro-Fuzzy Network to Forecast Electrical Load Time Series, IEEE Trans (2002)
12. Mutasem Alsmadi, Shahrul Azman Mohd Noah and Khairuddin Omar. Back Propagation Algorithm: The Best Algorithm Among the Multi-layer Perceptron Algorithm, <https://www.researchgate.net> (2009)
13. Mirza Cilimkovic, *Neural Networks and Back Propagation Algorithm*, Institute of Technology Blanchardstown (2010)
14. Heikki N.Koivo. *Neural Networks: Basics Using Matlab Neural Network Toolbox*, 59 (2008)
15. Mark Hudson Beale, Martin T.Hagan and Howard B.Demuth, *Neural Network Toolbox™ 7 User's Guide*, University of California, 951 (2011)