Leonardo Electronic Journal of Practices and Technologies
ISSN 1583-1078
Issue 16, January-June 2010
p. 119-128

Artificial Neural Network Application for Power Transfer Capability and Voltage Calculations in Multi-Area Power System

Palukuru NAGENDRA*, Sunita Halder nee DEY and Tanaya DUTTA

Department of Electrical Engineering, Jadavpur University, Kolkata-700032, West Bengal, India
E-mail: indrapn@gmail.com; sunitaju@yahoo.com; tanayadatta21@yahoo.co.in
(* Corresponding author: Phone: 09804353148, Fax: 03324146184)

Abstract

In this study, the use of artificial neural network (ANN) based model, multi-layer perceptron (MLP) network, to compute the transfer capabilities in a multi-area power system was explored. The input for the ANN is load status and the outputs are the transfer capability among the system areas, voltage magnitudes and voltage angles at concerned buses of the areas under consideration. The repeated power flow (RPF) method is used in this paper for calculating the power transfer capability, voltage magnitudes and voltage angles necessary for the generation of input-output patterns for training the proposed MLP neural network. Preliminary investigations on a three area 30-bus system reveal that the proposed model is computationally faster than the conventional method.

Keywords

Artificial neural networks; Multi-layer perceptron; Levenberg-Marquardt algorithm; Power transfer capability; Repeated power flow.

Major Symbols

P_r = real power interchange between areas	k = bus not in receiving area
P_{km} = tie line real power flow (from bus *k* in sending area to bus *m* in receiving area)	m = bus in receiving area
Y_{ij}, θ_{ij} = magnitude and angle of ij^{th} element of admittance of matrix Y	R = set of buses in receiving area
V_i, δ_i = magnitude and angle of voltage at i^{th} bus	n = set of all the buses
P_g, Q_g = real and reactive power outputs of generator	P_i, Q_i = net real and reactive powers at bus *i*
S_{ij} = apparent power flow through transmission line between bus *i* and bus *j*	

Introduction

Modern power systems are operating closer to their operating limits due to economic reasons and operational factors arising out of deregulation [1,2] and open market of electricity. Under such stressed conditions, the transfer capability becomes a major concern in system operation and planning [3, 4]. Power system transfer capability indicates how much inter area power transfers can be increased without compromising system security. Transfer capability computations are performed by the system operators to know the ability of the system to transfer power among areas within the system, and also by the system planners to indicate system's strength.

As the operating conditions of an interconnected power network vary continuously in real time, the power transfer capability of the network will also vary from instant to instant. For this reason, transfer capability and voltage calculations may need to be updated periodically for application in the operation of the network. In addition, depending on actual network conditions, transfer capabilities can often be different from those determined in the off-line studies. The most commonly used algorithms for computing power transfer capability are continuation power flow (CPF), optimal power flow (OPF) and repeated power flow methods [5, 6].

To give fast solutions to complex problems, some of which were hitherto revealed intractable by standard computing devices, artificial neural networks have recently been applied in different fields of research [7]. Many interesting ANN applications have been reported also in power system areas, where they are widely used in load forecasting, unit commitment, economic dispatch, security assessment, fault diagnosis and alarm processing [8]. Neural computing has attractive features, such as its ability to tackle new problems which are hard to define or difficult to solve analytically, its robustness in dealing with incomplete or fuzzy data, its processing speed, its flexibility and ease of maintenance.

In this paper, standard neural network architecture, multi-layer perceptron model for the computation of power transfer capabilities and voltages of multi-area power system has been proposed. The repeated power flow method, which repeatedly solves power flow equations at a succession of points along the specified load/generation increment, is used in this work for transfer capability and voltage calculations necessary for the generation of input-output patterns for training the proposed artificial neural network. The effectiveness of the ANN based approach is demonstrated on a three area 30-bus system for different loading patterns.

Conventional Repeated Power Flow Method

Referring to Fig.1, a simple interconnected power system is divided into three kinds of areas: receiving area, sending area and external areas. “Area” may be an individual electric system, power pool, control area, sub-regions, etc., which consist of a set of buses. The power transfer between two areas is the sum of the real powers flowing on all the lines which directly connect one area to the other.

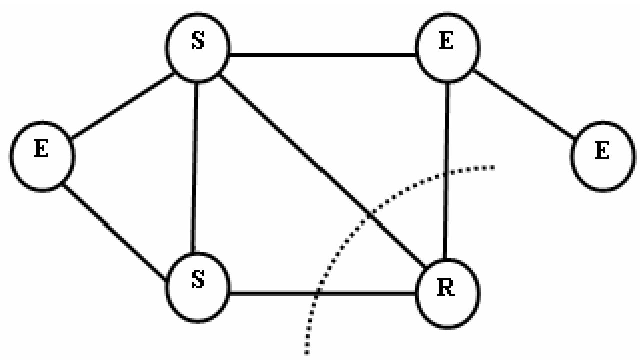

Figure 1. *A Simple Interconnected Power System*

The objective is to determine the maximum real power transfers from sending areas to receiving areas through the transfer path. In the mathematical formulation of the transfer capability computations problem, the following assumptions are made:

- The base case power flow of the system is feasible and corresponds to a stable operating point.
- The load and generation patterns vary very slowly so that the system transient stability is not jeopardized.
- The system has sufficient damping to keep within steady state stability limit.
- Bus voltage limits are reached before the system reaches the nose point and loses voltage stability.

The objective function to be optimized is

$$P_r = \sum_{m \in R, k \notin R} P_{km} \quad (1)$$

Subject to the power flow constraints given by

$$P_i = \sum_{j=1}^{n} |V_i||V_j||Y_{ij}|\cos(\theta_{ij} - \delta_i + \delta_j) \tag{2}$$

$$Q_i = -\sum_{j=1}^{n} |V_i||V_j||Y_{ij}|\sin(\theta_{ij} - \delta_i + \delta_j) \tag{3}$$

and the operational constraints

$$P_{g\ \min} \le P_g \le P_{g\ \max} \tag{4}$$

$$Q_{g\ \min} \le Q_g \le Q_{g\ \max} \tag{5}$$

$$S_{ij} \le S_{ij\ \max} \tag{6}$$

$$V_{i\ \min} \le V_i \le V_{i\ \max} \tag{7}$$

The control variables in this formulation are generator real and reactive power outputs, generator voltage settings, phase shifter angles, transformer taps and switching capacitors or reactors. The dependent variables are active and reactive power injections at slack bus, reactive power injection and bus voltage angle at generator buses.

The repeated power flow algorithm for the calculation of transfer capability is as follows.

1. Establish and solve the power flow problem for a base case.
2. Select a transfer case and solve for it.
3. Step increase in transfer power and solve for power flow problem.
4. Check for security limit violations of power flow through tie lines. If no violation, go back to step 3.
5. If there is any violation, decrease step size with minimum possible amount to eliminate them. This is the power transfer capability for the selected transfer case.

Multi-layer Perceptron Neural Network Model

Artificial neural networks were designed to mimic the characteristics of biological neurons in the human brain and nervous system. The network 'learns' by adjusting interconnections between layers. When the network is adequately trained, it is able to generalize relevant output for a set of input data. Learning typically occurs by example through training, where training algorithm iteratively adjust the connection weights (synapses).

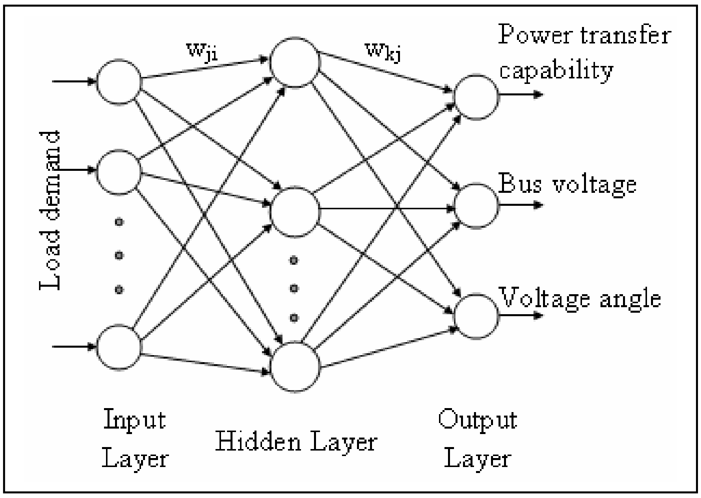

Figure 2. *Proposed MLP Structure with one Hidden Layer*

An MLP network consists of three layers: an input layer, an output layer, and one or more hidden layers [7]. Fig.2 illustrates a three-layered multi-layer perceptron network (MLPN). Each layer is composed of a predefined number of neurons. The neurons in the input layer only act as buffers for distributing the input signals x_i to neurons in the hidden layer. Each neuron j in the hidden layer sums up its input signals x after weighting them with the signals of the respective connections w_{ji} from the input layer, and computes its output y_j as a function f of the sum:

$$y_j = f\left(\sum w_{ji} x_i\right) \qquad (8)$$

where f is the activation function that is necessary to transform the weighted sum of all signals impinging onto a neuron. f is usually a sigmoidal or hyperbolic tangent function. The outputs of neurons in the output layer are computed similarly. Training a network consists of adjusting its weights using a training algorithm. In this paper the Levenberg-Marquardt (LM) algorithm [9] is used for training the MLP network. The LM algorithm is basically a Hessian based algorithm for nonlinear least square optimization. For neural network training the objective function is the error function of the type

$$\vec{e} = \sum_{k=1}^{p} \frac{1}{2}(t_k - y_k)^2 \qquad (9)$$

where y_k is the actual output for the k^{th} pattern and t_k is the desired output. p is the total number of training patterns.

The steps involved in training a neural network using LM algorithm are as follows:

i. Present all inputs to the network and compute the corresponding network outputs and errors. Compute the mean square error over all inputs as in (9).
ii. Compute the Jacobian matrix, *J(w)* where *w* represents the weights and biases of the network.
iii. Solve the Levenberg-Marquardt weight update equation to obtain Δw.
iv. Recompute the error using $w+\Delta w$. If this new error is smaller than that computed in step 1, then reduce the training parameter μ by μ^-, let $w= w+ \Delta w$, and go back the step1. If the error is not reduced, then increase μ by μ^+, and go back step 3.
v. The algorithm is assumed to have converged when the norm of the gradient is less than some predetermined value, or when the error has been reduced to some error goal.

The weights are updated according to the following formula:

$$w_{ji}\left(t+1\right) = w_{ji}\left(t\right) + \Delta w_{ji}\left(t\right) \quad (10)$$

with $$\Delta w_{ji} = \left[J^T(w)J(w) + \mu I\right]^{-1} J^T(w)E(w) \quad (11)$$

where E is a vector of size p calculated as

$$E = \left[t_1 - y_1 \;\; t_2 - y_2 \;\; ... \;\; t_p - y_p\right]^T \quad (12)$$

where $J^T(w)J(w)$ is referred as the Hessian matrix. I is the identity matrix, μ is the learning parameter.

Test System and Simulation Results

The proposed MLPN model is applied to a three area 30-bus system, the single line diagram of which is given in Fig. 3. The system data is given in appendix. The system is arbitrarily divided into three areas with 2 generators in each area. The power transfer capabilities between area 2 and area 3 are investigated for different loading conditions obtained by varying the active and reactive power loads in the system. The loads are varied with uniform power factor in such a way that the new load condition remains with in a range of 80 – 120% of the base operating condition of the system under consideration. In this study, using the RPF-based algorithm, the transfer capability from area 2 to area 3, bus voltage magnitudes and voltage angles in these areas are computed for different loading conditions.

This data is then used to train the ANN to provide real time evaluation of transfer capability, voltage magnitudes and voltage angles. Once the ANN is trained, the ANN ‘learns’ the implicit correlation between the loading patterns and the transfer capability patterns. Next, the new loading patterns (which have not been used to train the ANN) would be fed to the network and the network would provide the optimal power transfer capability, voltage magnitude and voltage angles at its output. The performance of the proposed MLPN method is presented in terms of relative error (ε), which is defined as

$$\varepsilon = \frac{|o_i - t_i|}{t_i} X\ 100\% \tag{13}$$

where t_i is the exact value from repeated power flow solutions and o_i is the output of ANN.

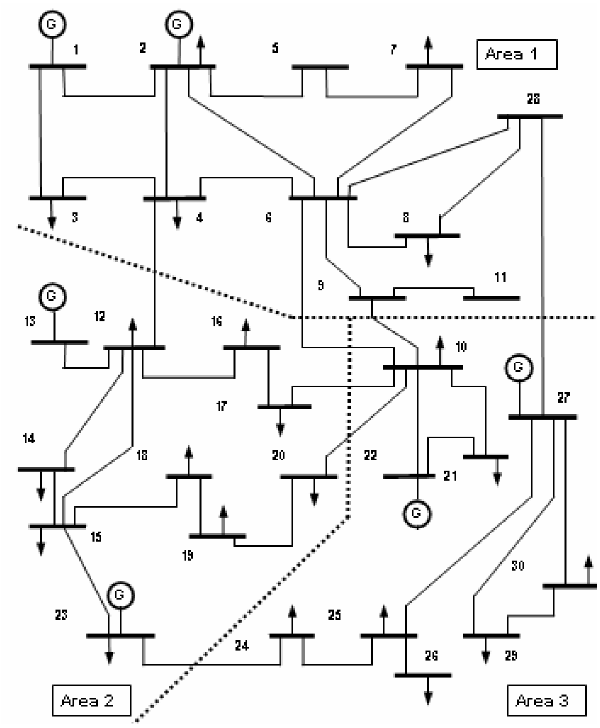

Figure 3. *Three area 30-bus system*

Table I shows the comparison of transfer capabilities from area 2 to area 3 obtained with proposed MLPN model against those obtained with the RPF method for different load operating conditions. The bus voltage magnitudes and voltage angles, calculated by the two methods for the power transfer capability case of area 2 to area 3 for 107.5% base operating condition are given in Table II. Fig. 5 and Fig. 6 show graphically the comparison of bus voltage magnitudes and voltage angles respectively calculated by the two methods for the transfer capability case of area 2 to area 3 for 112.5% base operating condition. From the simulation results, it can be seen that the proposed MLPN model is giving the results practically as accurate as that of conventional method. Further, it was observed that the proposed network with 16 inputs, 3 outputs and 9 neurons in the hidden layer takes only 0.94 second for an error goal of $1e^{-4}$, while the conventional method takes 2.81 seconds for the same computation.

Table 1. Transfer Capability from Area 2 to Area 3: Comparison of RPF and MLPN methods

Load Condition (%)	Transfer capability, MW		Relative error (%)
	RPF metho	MLPN metho	
92.5	26.4708	26.4746	0.0143
97.5	24.5007	24.5113	0.0432
102.5	23.1758	23.1868	0.0474
107.5	21.5064	21.5106	0.0195
112.5	19.8220	19.8236	0.0080
117.5	18.1225	18.1270	0.0248

Table 2. Voltage Magnitudes and Voltage Angles of Area 2 and Area 3 (107.5% Load Condition)

Bus no.	Bus Voltage magnitude, p.u.		Bus Voltage angle, deg.	
	RPF method	MLPN method	RPF method	MLPN method
10	0.9777	0.9776	-4.7099	-4.7077
12	0.9756	0.9756	-3.5361	-3.5341
13	1.0000	1.0000	-0.4925	-0.4905
14	0.9592	0.9592	-4.5277	-4.5257
15	0.9661	0.9661	-4.5333	-4.5312
16	0.9632	0.9630	-4.5184	-4.5163
17	0.9657	0.9655	-4.9607	-4.9584
18	0.9440	0.9440	-5.7293	-5.7271
19	0.9400	0.9400	-6.1094	-6.1071
20	0.9465	0.9465	-5.8632	-5.8610
21	0.9916	0.9917	-4.7017	-4.6993
22	1.0000	1.0000	-4.5653	-4.5628
23	1.0000	1.0000	-3.8722	-3.8699
24	0.9870	0.9870	-3.9179	-3.9154
25	0.9891	0.9891	-1.8301	-1.8272
26	0.9697	0.9697	-2.3150	-2.3120
27	1.0000	1.0000	-0.2429	-0.2398
29	0.9780	0.9780	-1.6428	-1.6321
30	0.9653	0.9653	-2.6284	-2.6248

Conclusions

In this paper, an artificial neural network model, multi-layer perceptron network has been developed for the computation of the power transfer capability among various areas and voltage magnitudes and voltage angles of the concerned buses of those areas in an interconnected system, accurately and rapidly for any loading conditions. Repeated power flow based transfer capability computation algorithm is utilized in generating the input-output patterns required for training the proposed ANN model. The preliminary investigations on a multi-area system indicate that the proposed model is computationally faster than the conventional RPF method and is useful for online applications.

References

1. William W.H., John F.K., *Electricity market restructuring: Reforms after reforms*, 20th Annual Conference Center for Research in Regulated Industries, May 2001.
2. Mala D., *Pricing of system security in deregulated environment*, Int. J Recent Trends in Engineering, 2009, 1(4), p. 49-51.
3. Ian D. et al., *Electric power transfer capability: Concepts, applications, sensitivity and uncertainty*, PSERC publication, 2001, p. 1-34.
4. Yan O., Singh C., *Assessment of available transfer capability and margins*, IEEE Trans. Power Systems, 2002, 17(2), p. 463-468.
5. Ajjarapu V., Chrity C., *The continuation power flow: A tool for steady state voltage stability analysis*, IEEE Trans. Power Systems, 1992, 17(1), p. 416-423.
6. Chakrabarti A., Sunita Halder, *Power system analysis: operation and control*, New Delhi, India, Prentice Hall of India (pvt) Ltd., 2nd edition, 2008.
7. Haykin S., *Neural Networks – A comprehensive foundation*, Prentice Hall Ltd, New York, 1999.
8. Vidya Sagar S.V., Rao N.D., *Artificial neural networks and their applications to power systems-a bibliographical survey*, Electric Power Systems Research, 1993, 28, p. 67-69.
9. Deepak M., Abhishek Yadav, Sudipta R., Prem K. K., *Levenberg-Marquardt learning algorithm for Integrate-and-Fire neuron model*, Neural Information Processing-Letters & Reviews, 2005, 9(2), 41-51.

Appendix

30-bus test system data is given below.

Table A1. Bus Data

SB	EB	R (p.u)	x (p.u)	b (p.u)	Line limit (MW)
1	2	0.02	0.06	0.03	130
1	3	0.05	0.19	0.02	130
2	4	0.06	0.17	0.02	65
3	4	0.01	0.04	0.00	130
2	5	0.05	0.20	0.02	130
2	6	0.06	0.18	0.02	65
4	6	0.01	0.04	0.00	90
5	7	0.05	0.12	0.01	70
6	7	0.03	0.08	0.01	130
6	8	0.01	0.04	0.00	32
6	9	0.00	0.21	0.00	65
6	10	0.00	0.56	0.00	32
9	11	0.00	0.21	0.00	65
9	10	0.00	0.11	0.00	65
4	12	0.00	0.26	0.00	65
12	13	0.00	0.14	0.00	65
12	14	0.12	0.26	0.00	32
12	15	0.07	0.13	0.00	32
12	16	0.09	0.20	0.00	32
14	15	0.22	0.20	0.00	16
16	17	0.08	0.19	0.00	16
15	18	0.11	0.22	0.00	16
18	19	0.06	0.13	0.00	16
19	20	0.03	0.07	0.00	32
10	20	0.09	0.21	0.00	32
10	17	0.03	0.08	0.00	32
10	21	0.03	0.07	0.00	32
10	22	0.07	0.15	0.00	32
21	22	0.01	0.02	0.00	32
15	23	0.10	0.20	0.00	16

Table A2. Load Data

Bus No.	P_d (MW)	Q_d (MVAR)
2	21.7	12.7
3	2.4	1.2
4	7.6	1.6
7	22.8	10.9
8	30.0	30.0
10	5.8	2.0
12	11.2	7.5
14	6.2	1.6
15	8.2	2.5
16	3.5	1.8
17	9.0	5.8
18	3.2	0.9
19	9.5	3.4
20	2.2	0.7
21	17.5	11.2
23	3.2	1.6
24	8.7	6.7
26	3.5	2.3
29	2.4	0.9
30	10.6	1.9

Table A3. Generator Data

Bus No.	P_g (MW)	Q_g (MVAR)	Q_{max} (MVAR)	Q_{min} (MVAR)	V_g (p.u)	P_{max} (MW)	P_{min} (MW)
1	23.54	0.00	150.0	-20.0	1	80	0
2	60.97	0.00	60.0	-20.0	1	80	0
22	21.59	0.00	62.5	-15.0	1	50	0
27	26.91	0.00	48.7	-15.0	1	55	0
23	19.20	0.00	40.0	-10.0	1	30	0
13	37.00	0.00	44.7	-15.0	1	40	0