Real Power Transfer Capability Calculations Using Multi-Layer Feed-Forward Neural Networks

X. Luo, *Student Member, IEEE*, A. D. Patton, *Life Fellow, IEEE*, and C. Singh, *Fellow, IEEE*

***Abstract*—This paper proposes a neural network solution methodology for the problem of real power transfer capability calculations. Based on the optimal power flow formulation of the problem, the inputs for the neural network are generator status, line status and load status and the output is the transfer capability. The Quickprop algorithm is used in the paper to train the neural network. A case study of IEEE 30-bus system is presented demonstrating the feasibility of this approach. The new method will be useful for reliability assessment in the new utility environment.**

***Index Terms*—Artificial neural network, Quickprop algorithm, Transfer capability, Optimal power flow, Reliability management.**

I. INTRODUCTION

ELECTRIC utilities around the world are confronted with restructuring, deregulation and privatization. In the environment of open transmission access, transmission networks tend to be more heavily loaded[1] and transmission service becomes one of the most critical elements. Transfer capability, acting as an indicator of the capability of a transmission network, is used by system operators to determine the ability to transfer power and system planners to indicate system's strength.

Power transfer distribution factors (PTDF), obtained by using a linear DC flow model, have been used to determine transfer capabilities [2],[3],[4]. This method is easy and computationally fast, but voltage and reactive power are not taken into consideration. These factors can, however, be very important in determining transfer capability.

A number of software tools, such as CPFLOW [8], TVLIM [9] and TRACE [10], have been developed for calculating transfer capability. Because of the uncertainty of the elements in the power system, there have been some trends toward probabilistic analysis of transfer capability [4]–[7]. A probabilistic distribution curve of transfer capability can be generated and the mean value and standard deviation of transfer capability can be calculated. This approach is useful for power system planning.

In recent years considerable progress has been achieved in the application of artificial neural networks to power system problems such as load forecasting, unit commitment, economic dispatch, security assessment, fault diagnosis and alarm processing [11]. 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.

A new model employing artificial neural networks to calculate transfer capability is developed in this paper. Based on the optimal power flow formulation for calculating real power transfer capability and the strong generalizing ability of the neural networks, the new model can calculate multi-area transfer capabilities quickly for a given power system status.

This paper is organized as follows. Section II provides a brief review of the multi-layer feed-forward neural network and the Quickprop algorithm, which is used in this paper to train this neural network. In section III, the problem of transfer capability calculations is formulated in terms of optimal power flow. The proposed methodology is implemented in section IV and a case study is given in section V to demonstrate the effectiveness of the presented method. Finally, a conclusion is made in section VI.

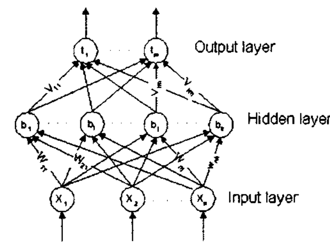

Fig. 1. Topology of a three layered MLP.

II. NEURAL NETWORK

A. Multi-Layer Feed-Forward Neural Network [13]

The multi-layer feed-forward neural network, also known as the multi-layer perceptron (MLP) network, was developed in the early 1970's and is the most popular topology in use today. A schematic diagram of the topology is given in Fig. 1. This network consists of a set of n input neurons, m output neurons and one or more hidden layers of k intermediate neurons. Data flows into the network through the input layer, passes through the hidden layers and finally flows out of the network through the output layer. The network thus has a simple interpretation as a form of input-output model, with network weights as free parameters. Such networks can model functions of almost any arbitrary complexity, with the number of layers and the number of neurons in each layer, determining the function complexity.

In Fig. 1, the input signals X_i $(i = 1, ..., n)$ are multiplied by the weights W_{ij}; then operated on by the activation function

Manuscript received December 14, 1998. This work was supported by NSF Grant ECS-9412712 and Energy Resources Grant 95-20.

The authors are with the Department of Electrical Engineering, Texas A&M University, College Station, TX 77843 USA.

Publisher Item Identifier S 0885-8950(00)03829-3.

$f(x)$ to produce the b_j of the hidden layer. Similar operations can be made on output layer neurons to produce the outputs of the network. Here

$$b_j = f(\sum_{i=1}^{n} X_i W_{ij}) \tag{1}$$

where f is a transfer function or activation function which can take the form of a sigmoid function, hyperbolic tangent function, piecewise linear function or threshold function, depending on the problem to be solved. The sigmoid function is commonly used as the following form:

$$f(x) = (1 + e^{-x})^{-1} \tag{2}$$

MLP operates in two modes: training and testing. Training is a procedure used to minimize the difference between outputs of MLP and the desired values by adjusting the weights of the network. Sets of input vectors are presented to the network until training is completed. Then the network's weights are "frozen" in the trained state and the new input data are presented to the network to determine the appropriate output.

B. Quickprop Algorithm

Most neural networks use some form of the back-propagation algorithm [13]. However, back-propagation is too slow for many applications. When a neuron with a standard sigmoid function is near its boundary, the slope of the activation function is almost zero and error gradient is near zero. As a result, the update weight step will be very small.

The Quickprop algorithm, which was put forward by Scott Fahlman in 1988 [12], is a modified back-propagation algorithm developed to speed up the training of the network. The Quickprop algorithm, which uses a heuristic second-order optimization technique somewhat similar to Newton's method, is based on two assumptions about the weight space: (1) the weights can be adjusted independently, (2) if the error of the outputs is plotted as a function of any given weight it will form an upward-opening parabola. It is an interactive technique because these assumptions do not hold in a typical network.

The weight update equation for Quickprop algorithm is:

$$\Delta W_{ji}(t) = \frac{g_{ji}(t)}{g_{ji}(t-1) - g_{ji}(t)} \Delta W_{ji}(t-1) \tag{3}$$

where:

$g_{ji}(t) = \partial E / \partial W_{ji}(t)$,	partial derivative of the error function by W_{ji}
$E = \frac{1}{2}\sum_{i=1}^{N}\sum_{m=1}^{M}(t_i^m - r_i^m)$,	mean square error
N	number of training patterns
M	number of output neurons
t	desired output values
r	computed neural network output values

When the step computed by the Quickprop formula is too large, one parameter called "mu" (maximum growth factor) is used to limit the size of the weight change to less than or equal to mu times the previous weight change.

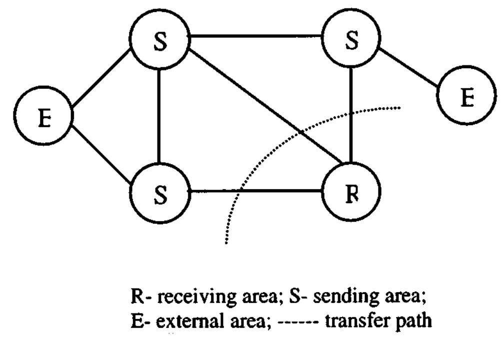

Fig. 2. A simple interconnected power system.

III. Problem Formulation

Referring to Fig. 2, a simple interconnected power system can be divided into three kinds of areas: receiving area, sending areas and external areas. "Area" can be defined in an arbitrary fashion. It may be an individual electric system, power pool, control area, subregions, etc. The objective is to determine the maximum real power transfers from sending areas to receiving area through the transfer path.

The physical and electrical characteristics of the systems limiting the transfer capability include:

- Thermal limits - constrain the amount of transfer that transmission line can safely handle without overload.
- Voltage limits - voltages over the transmission system should be within acceptable operation ranges.
- Stability limits - voltage stability and angle stability must be maintained.

The calculation of transfer capability can be formulated as an optimal power flow as follows:

Maximize

$$P_r = \sum_{m \in R, k \notin R} P_{km} \tag{4}$$

Subject to:

$$P_i - \sum_{j \in N} V_i V_j Y_{ij} \cos(\theta_{ij} + \delta_j - \delta_i) = 0 \tag{5}$$

$$Q_i - \sum_{j \in N} V_i V_j Y_{ij} \sin(\theta_{ij} + \delta_j - \delta_i) = 0 \tag{6}$$

$$P_{g,\min} \le P_g \le P_{g,\max} \tag{7}$$

$$Q_{g,\min} \le Q_g \le Q_{g,\max} \tag{8}$$

$$S_{ij} \le S_{ij,\max} \tag{9}$$

$$V_{i,\min} \le V_i \le V_{i,\max} \tag{10}$$

where:

P_r	the interchange real power sending areas to receiving area
k	bus not in receiving area
m	bus in receiving area
P_{km}	tie line real power flow (from bus k in sending areas to bus m in receiving area)
R	set of buses in receiving area

N	set of all the buses
Y_{ij}, θ_{ij}	magnitude and angle of ijth element of the admittance matrix Y
V_i, δ_i	magnitude and angle of voltage at bus i
P_g, Q_g	real power and reactive power output of generation
P_i, Q_i	net real power and reactive power at bus i
S_{ij}	apparent power flow of transmission line
$P_{g,\min}, P_{g,\max}$	minimum and maximum real power output of generator g
$Q_{g,\min}, Q_{g,\max}$	minimum and maximum reactive power output of generator g
$V_{i,\min}, V_{i,\max}$	minimum and maximum of voltage magnitude at bus i
$S_{ij,\max}$	maximum allowed appraent power flow

The control variables in the above formulation are generator real and reactive power output, generator voltage settings, phase shifter angles, transformer taps and switchable capacitors or reactors. The dependent variables in the above formulation are slack bus real and reactive power, PV bus reactive power and voltage angle, PQ bus voltage magnitude and voltage angle.

An optimal power flow algorithm can be used here to solve above optimization problem and get the maximum transfer capability between areas for the specified system condition.

IV. Implementation Algorithm

From the formulation in Section III, we can see that transfer capability is a complex nonlinear function of customer demand, system topology and generation availability. A neural network approach to solve the transfer capability problem is presented in this section.

1) Input Vector: Generation status, load level and line status define a specified power system state. Therefore the input vector consists of following three parts:

- Generation status
  - 1 - generator is available
  - 0 - generator is unavailable
- Line status
  - 1 - line is available
  - 0 - line is unavailable
- Load conditions

It is assumed that each bus changes its load at the same rate within the area, but the rate may differ for different areas. The number of input neurons representative of load conditions is equal to the number of system areas. For each area, if the load is equal to base load, the input is 1.0; if the load is 110% of the base load, the input value is 1.1, etc.

For a large power system, since the number of generators and lines is large, it is important to find those critical generators and lines whose unavailability will have the largest effect on transfer capability. Contingency screening and ranking techniques are used to find those critical generators and lines. Only the status of these elements are taken as inputs, thus the number of input neurons can be reduced, which is advantageous for the training of the neural network.

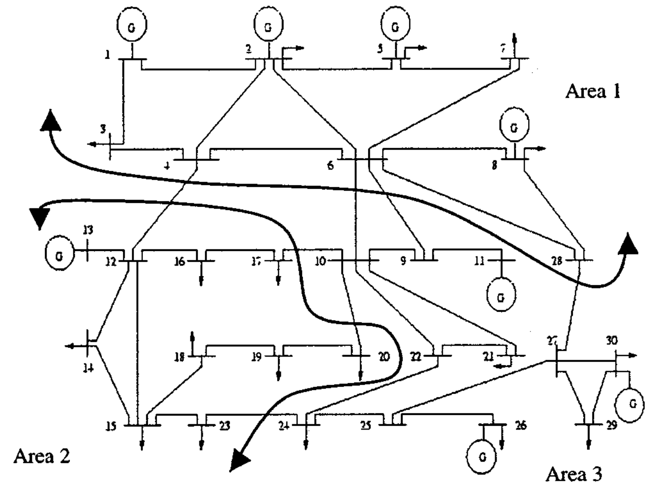

Fig. 3. Single line diagram of the modified IEEE-30 bus system.

2) Output Vector: Only one output signal is used here, the transfer capability between the sending areas and the receiving area.

3) Network Architecture and Training: The complexity of a neural network is characterized by the number of hidden layers and the number of neurons in each hidden layer. There is no general rule for selection of these parameters. The critical issue for developing a neural network is generalization. Like other nonlinear estimation methods such as kernel regression, neural networks can suffer from either underfitting or overfitting. A neural network with a small number of neurons may not be sufficiently powerful to model a complex function. For example, a network with no hidden layers actually models a simple linear function. On the other side, a neural network with too many neurons may lead to overfitting the training sets and lose its ability to generalize which is the main desired characteristic of a neural network. In [14], five effective ways to avoid overfitting are: model selection, jittering, weight decay, early stopping and Bayesian estimation. Here we will use "early stopping" in the training process as a means of avoiding overfitting.

V. Studies

A. Test System

For the purpose of testing, the proposed method was applied to a modified IEEE 30-bus system. In order to study more generator outages, the basic IEEE 30-bus system was modified by adding two generators to bus 26 and 30, respectively, where each unit has a maximum capacity of 30MW. The modified system includes 8 generators, 21 load buses and 41 transmission lines. Data of the original IEEE 30-bus system can be accessed from URL address: ftp://wahoo.ee.washington.edu. Single line diagram of this system is shown in Fig. 3. The system is divided into 3 areas and the transfer capability to be calculated is from areas 1 and 3 to area 2.

B. Training Patterns of the Neural Network

Training sets provided to the neural network are representative of the whole state space of concern so that the trained neural network has the ability of generalization. We assume only one

TABLE I
AVERAGE ERROR COMPARISON OF SIX DIFFERENT ARCHITECTURES (err/unit)

MLP structure	# of iteration:5000 Training	testing	# of iteration:10000 training	testing	# of iteration:15000 training	testing
22/15/5/1	0.02018	0.02961	0.01669	0.02795	0.01525	0.02865
22/20/5/1	0.02400	0.03486	0.01991	0.03329	0.01829	0.03198
22/25/5/1	0.01724	0.02856	0.01508	0.02785	0.01401	0.02780
22/30/5/1	0.03593	0.04731	0.02984	0.04126	0.02722	0.03878
22/25/6/1	0.03285	0.04420	0.02732	0.04038	0.02413	0.03852
22/25/4/1	0.02329	0.00333	0.01707	0.02895	0.01441	0.02785

line is on outage at a time because the outage probability of a line is very small. The outage probabilities of generators are larger than those of transmission lines, so we assume that it is possible for two generators to be on outage at the same time. Load levels vary from 50% to 150% of base load.

Training patterns for the IEEE 30-bus system are composed of:

- Load levels for each area from 50% to 150% of base load while all lines and generators remain in operation.
- Generator outages (including one and two generators on outage) at 75%, 90%, 100%, 115%, 125% of the base load of area 2.
- Single line outages at 75%, 90%, 100%, 115%, 125% of the base load of area 2.
- Joint outage (one generator and one line) at 75%, 90%, 100% and 125% of the base load of area 2.

There are 330 training patterns in total. This may not be an ideal set of training patterns, but it covers the range of load levels and the outage of generators and transmission lines.

C. Test Patterns of the Neural Network

The trained neural network was tested using 130 test cases which are composed of load variations and generator and line outages. None of these test cases were used in the training of the neural network.

D. Architecture of the Neural Network

1) Input Layer: The input layer is composed of the neurons which are representative of the load conditions, generator and line status. For this modified IEEE 30-bus system, 3 input neurons are taken to represent the load conditions in each of the 3 areas and 8 neurons are taken for the status of each of the 8 generators. Since there are a total of 41 lines, a line contingency screening technique was used to find those critical lines which have the greatest effect on maximum transfer capability. Here we identified 11 lines as critical: 4-12, 6-28, 8-28, 6-10, 9-10,10-17, 12-15, 22-24, 23-24, 25- 27, 27-28. Thus 11 input neurons represent the critical line status. The total number of input neurons is thus 22.

2) Output Layer: The output layer has only one neuron here, whose output is the transfer capability from area 1 and 3 to area 2.

3) Data Scaling: Scaling either input or target variables tends to make the training process better behaved by improving the numerical condition of the optimization problem and

TABLE II
NEURAL NETWORK STATISTICS

Input neurons	22
Output neuron	1
Neurons in hidden layer 1	25
Neurons in hidden layer 2	5
Training patterns	330
Testing patterns	130
Mu parameter	1.75
Learning rate	0.001
Total iteration number	20000
Training (err/unit)	0.01395
Testing (err/unit)	0.02775

ensuring that various default values involved in initialization and termination are appropriate. Here in our study, the values of the input vectors are between 0 and 1.5, thus there is no need to scale them. The output vector, which is the value of transfer capability, varies a lot. Therefore, we scale the output value and make it between 0 and 1.

4) Network Topology: A neural network with one hidden layer was tried first, but was found hard to converge. Therefore, a neural network with two hidden layers was selected for further analysis. Table I shows the average error at different numbers of iterations for six different neural network topologies. The Table I Average error comparison of six different architectures (err/unit) structure 22/15/5/1 means that there are 22 input neurons, 15 neurons in the first hidden layer, 5 neurons in the second hidden layer and one neuron for output. From Table I, we can see the structure 22/25/5/1 converges quickly and is more accurate than the others. Hence we have chosen it as the network architecture in our example.

Sigmoid transfer functions are used for the hidden layer and a linear transfer function is used for the output layer.

E. Analysis of Results

Based on the selected training and testing patterns and the chosen neural network topology, the Quickprop algorithm is used to train the neural network. Table II shows the training and testing statistics for the chosen neural network as applied to the test system.

Tables III.1-III.3 give the relative error list of OPF outputs(exact transfer capability) and neural network output (approximate transfer capability) for different test cases. Fig. 4 shows graphically the neural network estimates for transfer

TABLE III.1
RELATIVE ERROR LIST OF LOAD VARIATION TEST CASES

case	relative error	case	relative error	case	Relative error	case	Relative error
1	1.12%	8	1.40%	15	0.11%	22	0.49%
2	0.55%	9	4.65%	16	0.11%	23	0.72%
3	0.76%	10	0.52%	17	0.12%	24	0.20%
4	1.11%	11	0.11%	18	0.12%	25	0.63%
5	0.81%	12	0.08%	19	0.11%	26	0.83%
6	1.96%	13	0.01%	20	0.12%		
7	2.86%	14	0.11%	21	0.02%		

*Case 1-10: Area 2 loads vary while areas 1&3 loads are constant at the base values
*Case 11-20: Area 1 loads vary while areas 2&3 loads are constant at the base values
*Case 21-26: Area 3 loads vary while area 1&2 loads are constant at the base values

TABLE III.2
RELATIVE ERROR LIST OF SINGLE OUTAGE TEST CASES

case	relative error	case	relative error	case	relative error	case	relative error
27	1.18%	40	1.51%	53	4.81%	66	1.86%
28	0.10%	41	6.12%	54	2.42%	67	2.24%
29	3.84%	42	1.46%	55	2.63%	68	3.69%
30	0.15%	43	1.11%	56	0.55%	69	4.12%
31	2.72%	44	2.30%	57	8.01%	70	2.46%
32	2.27%	45	4.81%	58	1.09%	71	3.83%
33	5.71%	46	2.42%	59	1.46%	72	0.92%
34	2.54%	47	2.63%	60	10.59%	73	1.84%
35	0.93%	48	0.55%	61	1.26%	74	5.47%
36	0.00%	49	8.01%	62	0.39%	75	0.32%
37	0.04%	50	1.09%	63	4.34%	76	1.62%
38	6.62%	51	1.46%	64	0.34%	77	2.01%
39	0.93%	52	10.59%	65	3.30%	78	9.12%

*Case 27-45: Generator outages with area 2 loads at 110% of the base and areas 1&3 loads at base values
*Case 46-54: Line outages with area 2 loads at 110% of the base and areas 1&3 loads at base values
*Case 55-61: Line outages with area 2 loads at 65% of the base and areas 1&3 loads at base values
*Case 62-78: Generator outages with area 2 loads at 65% of the base and areas 1&3 loads at base values

TABLE III.3
RELATIVE ERROR LIST OF JOINT OUTAGE TEST CASES

case	relative error	case	relative error	case	relative error	case	relative error
79	0.88%	92	2.02%	105	2.73%	118	2.15%
80	17.4%	93	0.23%	106	4.28%	119	5.64%
81	2.55%	94	0.78%	107	1.94%	120	1.63%
82	1.09%	95	3.40%	108	19.3%	121	2.25%
83	0.07%	96	1.39%	109	21.2%	122	3.82%
84	2.96%	97	0.89%	110	12.4%	123	4.26%
85	0.16%	98	4.22%	111	0.55%	124	5.99%
86	1.42%	99	0.34%	112	1.58%	125	1.55%
87	2.47%	100	0.34%	113	4.56%	126	3.03%
88	0.20%	101	13.2%	114	0.84%	127	2.32%
89	0.35%	102	13.6%	115	0.68%	128	12.37%
90	0.35%	103	2.48%	116	1.84%	129	4.91%
91	25.7%	104	1.28%	117	1.98%	130	14.40%

*Case 79-110 joint outage with area 2 loads at 110% of the base and area 1&3 loads at base values
*Case 111-130 joint outage with area 2 loads at 65% of the base and area 1&3 loads at base values

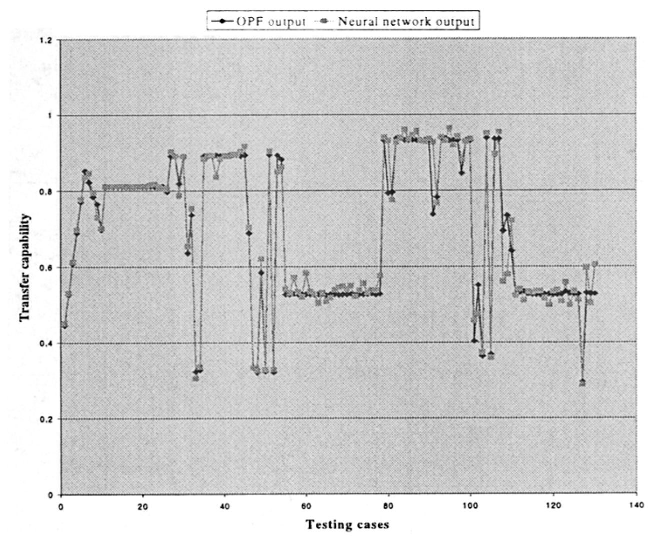

Fig. 4. Scatter plot of neural network output and OPF output.

capability as compared to exact values as determined from OPF calculations. The base value in Fig. 4 is 70MW.

Relative error is defined as follows:

$$\text{Relative Error} = \frac{|r_i - t_i|}{t_i} * 100\% \tag{11}$$

where:

t_i is the exact value from OPF

r_i is the output of neural network

Tables III.1-III.3 show that of the total 130 testing patterns, 112 are within an error of 5%, 8 errors are between 5%-10%, 6 errors are between 10%-15% and 4 cases have errors greater than 15%. The greatest error occurred in case 91(joint outage of G11 and line 9-10 at 110% of base load in area 2) where the error is 25.75%. Also from Fig. 4 and Tables III.1-III.3, we can see that for the load variation test cases (cases 1-26), neural network results are very close to those of OPF results. This indicates that the neural network can accurately estimate transfer capabilities for varying load levels. Similarly, errors are small for test cases 27-78 in which generator outages only or line outages only were studied. Thus, the neural network can also accurately estimate transfer capabilities for varying generator or line status. Errors are seen to be larger, though still generally acceptable, in test cases 79-130 where joint outage of generator and line were studied.

F. Comparison of Quickprop and Delta-Bar-Delta Training Algorithm

Delat-Bar-Delta is another popular and improved Back-propagation learning algorithm for MLP, which use a learning method where each weight has its own self-adapting coefficient. The connection weights are updated on the basis of the partial derivatives of the error with respect to the weights as following:

$$W_{ji}(t) = W_{ji}(t-1) - \alpha_{ji}(t) * \frac{\partial E}{\partial W_{ji}} \tag{12}$$

Where:

TABLE IV
AVERAGE ERROR COMPARISON OF DELTA-BAR-DELTA AND QUICKPROP ALGORITHM (err/unit)

Learning algorithm	# of iteration:5000 training	# of iteration:5000 testing	# of iteration:10000 training	# of iteration:10000 testing
Delta-Bar-Delta	0.09949	0.10721	0.06281	0.06471
Quickprop	0.01724	0.02856	0.01508	0.02785

$W_{ji}(t)$ the weight from neuron j to neuron i at time t.
$\alpha_{ji}(t)$ the learning rate of the weight from neuron j to neuron i at time t.
$\partial E/\partial W_{ji}$ the partial derivative of the error function with respect to W_{ji}

Table IV is a comparison of the Quickprop and Delat-Bar-Dela algorithms on the network architecture 22/25/5/1 for the same training and testing patterns used before. From the results in Table IV, we can see that Quickprop algorithm converges faster than the Delta-Bar-Delta algorithm.

VI. CONCLUSION

The transfer capability estimation method proposed in this paper is capable of reflecting variations in load levels and in the status of generation and transmission lines. Using the IEEE 30-bus system, the method is shown to accurately estimate transfer capabilities between system areas with variations in load levels, in the status of generation, and in the status of lines.

Quickprop algorithm is used to train the neural network in this paper. There are, however, other training algorithms such as RPROP, SuperSAB, SASS, and Conjugate Gradient. It is hard to say which one is better in general. For different problems, different algorithms may be chosen to train the neural network.

Though the chosen neural network architecture seems to work well in this paper, we do not claim it is optimal. Indeed, the process of choosing the number of hidden layers, the number of neurons for each layer, how to deal with overfitting and underfitting during the training procedure, remains an open research topic. Also which physical variables are chosen as input variables to the neural network and how many cases to use in the training patterns depend on experience and the availability of the data.

We believe that the proposed method may have important applications in power system operation, planning and reliability assessment. The method would allow a system operator to immediately update transfer capabilities as loads and the statuses of generation units and transmission lines change. This should enhance the economy and security of a system. Similarly, because the method can very quickly estimate transfer capabilities when system conditions change, the method should be useful in planning and reliability studies where a wide range of system conditions must be considered and evaluated.

The proposed method can also be used for Available Transfer Capability (ATC) calculation under the open access environment.

REFERENCES

[1] A. D. Patton, C. Singh, and D. G. Robinson, "The Impact of Restructuring Policy Changes on Power Grid Reliability,", Final Report, SAND98-2178, Oct. 1998.

[2] G. L. Landgren, H. L. Terhune, and R.K. Angel, "Transmission Interchange Capability- Analysis by Computer," *IEEE Transactions on Power Apparatus and System*, vol. PAS-91, no. 6, Nov/Dec 1972.

[3] G. L. Landgren and S. W. Anderson, "Simultaneous Interchange Capacity Calculations," *IEEE Transaction on Power Apparatus and Systems*, vol. PAS-92, no. 6, Nov/Dec. 1973.

[4] G. T. Heydt and B. M. Katz, "A Stochastic Model in Simultaneous Interchange Capacity Calculations," *IEEE Transactions on Power Apparatus and Systems*, vol. PAS-94, no. 2, March/April 1975.

[5] F. Xia and A. P. S. Meliopoulous, "A Methodology for Probabilistic Simultaneous Transfer Capability Analysis," in *1995 IEEE PES Summer Meeting*, July 1995, Paper 95 SM 5140-0 PWRS.

[6] G. Hamoud, "probabilistic Assessment of Interconnection Assistance Between Power Systems," *IEEE Transaction on Power System*, vol. 13, no. 2, May 1997.

[7] J.C.O. Mello, A.C.G. Melo, and S. Granville, "Simultaneous Transfer Capability Assessment by Combining Interior Point Methods and Monte Carlo Simulation," *IEEE Transaction on Power System*, vol. 12, no. 2, May 1997.

[8] A. J. Flueck, H.-D. Chiang, and K. S. Shah, "Investigation the Installed Real Power Transfer Capability of a large Scale Power System Under a Proposed Multiarea Interchange Schedule Using CPFLOW," *IEEE Transaction on Power System*, vol. 11, no. 2, May 1996.

[9] S. Wunderlich, T. Wu, R. Fischl, and R. O'Connell, "An Inter-Area Transmission and Voltage Limitation (TVLIM) Program," *IEEE Transaction on Power System*, vol. 10, no. 3, August 1995.

[10] EPRI, "Solving the Transfer Capability Puzzle (A Newsletter from the Power Delivery Group),", September 1997.

[11] T. S. Dillon and D. Niebur, *Neural Networks Applications in Power Systems*. London: CRL Publishing Ltd, 1996.

[12] S. E. Fahiman, "An Empirical Study of Learning Speed in Back-Propagation Networks," Carnegie-Mellon University, Computer Science, Tech Report CMU-CS-88- 162, Sept. 1988.

[13] L. Fausett, *Fundamentals of Neural Networks*: Prentice-Hall, Inc., 1994.

[14] , ftp://ftp.sas.com/pub/neural/.

X. Luo received his BS and MS degree in EE from Shanghai Jiao Tong University in July, 1992 and February 1995 respectively. He then worked for nearly 2 years in East China Electric Power Design Institute, Shanghai, China, mainly in power system design and planning. He is currently a Ph.D. student in the EE Dept. at Texas A&M University. His research interests include power system reliability, application of neural networks in power system.

A. D. Patton received the Ph.D. degree from Texas A&M University in 1972, where he is presently Professor of EE. He was an engineer in the Electric Utility Engineering Department of Westinghouse Electric Corp. during 1957-1965 and has been president of Associate Power Analysts, Inc. since 1973. His research interests include power system reliability evaluation and the application of neural networks to power system problems. Dr. Patton is a life Fellow of the IEEE, a member of Eta Kappa Nu, Tau Beta Pi, Sigma Xi, C.I.G.R.E., National Society of Professional Engineers, and a registered professional engineer in the state of Texas.

C. Singh is Professor and Department Head of Electrical Engineering, Texas A&M University. He received the 1986-87 Haliburton Professorship, and the 1992-1993 Dresser Professorship. He also served as Director, NSF Power System Program, for the year 1995-96. Dr. Singh is a senior TEES Fellow at Texas A&M University and recipient of the IEEE 1998 Distinguished Power Engineering Educator Award.