ISSN 2278 - 8875
International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering
Vol. 1, Issue 6, December 2012

Load Flow Analysis Using Particle Swarm Optimization Trained Neural Network

M.Ulagammai

Assistant Prof (SG), Saveetha Engineering College, Chennai, India

ABSTRACT: A new method of load flow analysis of a power system using Particle Swarm Optimization (PSO) trained neural network is proposed in this paper. Load Flow Analysis is the first and foremost step in power system analysis. It is necessary to solve load flow problem for Power System Planning, Contingency analysis, and Stability analysis. The results obtained from the load flow analysis are Voltage magnitudes and phase angles at various buses and line losses. Newton-Raphson (NR) method is mostly used for solving the load flow problem. This conventional method involves the formation of Jacobian matrix, which increases the complexity of the solution algorithm, and also it fails to solve the problem for ill conditioned systems where line reactance to resistance ratio is small.

The proposed method using PSO trained neural network reduces the mathematical computations. Trained neural network is used for solving the load flow problem. The generalized back propagation algorithm for training the neural network is replaced by an optimization technique. The weighing factors in the weighing nodes of the neural network constructed are tuned using PSO. With the advantage of global search abilities of PSO, the network constructed results in the better training. The above-proposed method is tested on standard 5 bus system. The results are compatible with the results obtained by conventional NR method.

Nomenclature:

n- number of buses
V_i- Voltage magnitude at bus i
δ_i – Voltage angle at bus i in radians
Y_{ij} – Magnitude of ij^{th} element of bus admittance matrix
θ_{ij} – Angle of ij^{th} element of bus admittance matrix
P_{gi} – Real power generation at bus i
Q_{gi} - Reactive power generation at bus i
P_{di}- Real power demand at bus i
Q_{di} - Reactive power demand at bus i

I. INTRODUCTION

The first and foremost step in solving any problem in power system is to conduct load flow test for that power system. Load flow studies are necessary for planning, operation, economic scheduling, and for many other analysis such as transient stability, voltage stability and contingency studies. The results obtained from the load flow studies are essential for continuous monitoring of the current state of the system. Since the power flow equations are non linear, the problem of load flow is solved using numerical techniques.

Conventional methods of solving load flow problem are

1. Gauss Seidel method
2. Newton Raphson method
3. FDLF method

The most widely used method is Newton Raphson method. Since the solution is obtained by taking the inverse of Jacobian matrix, the conventional methods fail to solve the load flow problem in some cases (Ill conditioned systems).

This paper proposes an evolutionary algorithm for solving the load flow problem using PSO trained neural network. The PSO was used in some applications in the power system field such as the solution of OPF or the Reactive Power Planning problem [3]. Hybrid PSO technique has been applied to solve the balanced load flow problem [4] Similarly Neural Network is used to solve the load flow problem. But training of neural network is a big problem. To overcome this neural network is trained using PSO algorithm. The weights and biases of the network are tuned using this PSO technique whose objective is the minimization of error and the trained neural network is used for solving the load flow problem.

Load flow Equation:

The real Power at any bus i is given by

$$P_{ci} = V_i \sum_{j=1}^{n} V_j Y_{ij} \cos(\theta_{ij} - \delta_i + \delta_j) \quad \text{--1}$$

Similarly the reactive power in polar form is

$$Q_{ci} = -V_i \sum_{j=1}^{n} V_j Y_{ij} \sin(\theta_{ij} - \delta_i + \delta_j) \quad \text{--2}$$

II. ARTIFICIAL NEURAL NETWORKS

The artificial neural network (ANN) is about a network of interconnected elements.In general, the function of a ANN is to produce an output pattern when presented with an input pattern. The feed-forward ANNs are commonly trained in a supervised fashion with the error back-propagation (BP) algorithm. The algorithm consists of a forward pass and a backward pass. It is applied by first passing an input signal forward through the network in the direction towards the output until a set of the actual response is obtained from the network. The error signal is then generated based on the difference between the actual and the target response. Finally the generated error signal is passed backwards through the hidden layers, in the direction towards the inputs. During the forward pass, the synaptic weights of the network are fixed. It is during the backward pass that the synaptic weights are adjusted to adapt the network in producing desired outputs.

The basic BP algorithm is a gradient descent algorithm, which adjusts the network weights along the steepest descent direction of the error function (that is, the direction in which the error function decreases most rapidly; negative of the gradient).

The weight adjustment for the BP algorithm is given by,

$$W_k^p(t+1) = W_k^p(t) + m\delta_k^p x_t^p + \alpha \Delta W_k^p(t) \quad \text{(Num)}$$

$$b_k^p(t+1) = b_k^p(t) + m\delta_k^p \quad \text{(Num)}$$

Where x is the input vector, m is the learning rate parameter, α is the momentum bias coefficient term ranging from 0 to 1 and δ is the negative derivative of the total squared error in respect to the neuron's output, *k* and *p* are iteration count and pattern count respectively. The momentum term helps to smooth the local curvature between successive squared error surfaces and thereby improves convergence characteristic of the BP learning algorithm.

In general the weights and biases are updated by means of delta rule as per the equation (NUMBERS). To have better tuning of weights and biases and to minimize the error between calculated and actual output, an evolutionary computation technique PSO is introduced.

III. PARTICLE SWARM OPTIMIZATION

The particle swarm optimization algorithm was introduced by Kennedy and Eberhart in 1995. The PSO was inspired by insect swarms and is motivated from the simulation of social behavior instead of the evolution of nature as in the evolutionary algorithms (genetic algorithms). It is a population based algorithm and has since proven to be a powerful tool for optimization problem The algorithm is also very simple. The PSO model consists of a number of particles moving around in the search space, each representing a possible solution to a numerical problem. Each particle has a position vector (x_i) and a velocity vector (**y_i**), each particle keeps track of its coordinates in the problem space, which are associated with the best solution (fitness) it has achieved so far. This value is called Pbest. Another best value that is tracked by the global version of the particle swarm optimizer is the over all best value, and its location, obtained so far by any particle in the population. This location is called gbest.

The PSO concept consists of, at each time, changing the velocity of each particle flying towards its pbest and gbest. Acceleration is weighted by random numbers. The velocity vector is given by

ve= v* w +a (gbest-x) + b(pbest-x) -----4

In each iteration the position of each particle is updated by using this velocity vector by

x=x + ve -------5

where a &b are random numbers & w is weight factor.

To improve the performance and for faster convergence two modifications are done in the PSO algorithm

ISSN 2278 - 8875
International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering
Vol. 1, Issue 6, December 2012

1. Random numbers a &b are varying between 0 &2 if the no of iterations is lesser than 1000. For the iterations greater than 1000 the random values are varying between 2&3.
2. The worst position is replaced by the average of the best positions.

IV.PROPOSED ALGORITHM FOR LOAD FLOW PROBLEM

1. Data for training the neural network are obtained from the NR algorithm

(The inputs are the slack bus voltage, Generator bus voltage and real power and the Demands at load buses. The outputs are bus voltage magnitude and angle at various buses.)

2. The inputs are normalized between 0 and 1
3. PSO Training:

The input and output neurons of the network constructed depends on the number of buses in the system. The weights of the connecting layers are obtained using the PSO algorithm.

i) The objective function is to minimize the error between the calculated output and actual output.(Calculated output is the output from the neural network and actual output is the data obtained from NR method)

ii) The number of optimizing variables depends on input neurons (n_i), hidden neurons (n_h) and output neurons (n_o) and their connecting layer weights. The connection strength of the network is optimized by the above said PSO algorithm to reduce the error between the calculated and actual output.

iii) After certain iteration, the weights are taken as the output.

4. Simulation: Using the trained neural network, the output is obtained by simulating the input patterns. After simulation, the neural network output is descaled to generate the desired forecasting loads.
5. This Neural network is used for solving the load flow problem. ie., The inputs are varied according to the load demand. The outputs, the voltage magnitudes and angles at various buses, are directly obtained from the neural network.

V. RESULTS AND DISCUSSIONS

The proposed algorithm is tested on sample 5 bus system [1].

Fig 1 shows the one line diagram of the sample system. Table 1 shows the line data. Table 2 shows the real and reactive power demand for the balanced system.

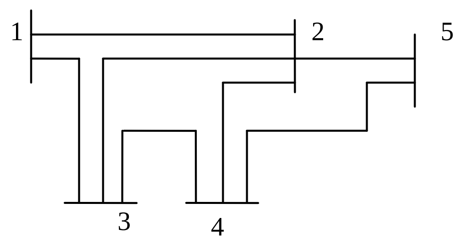

Table 1: Line Data

Bus code	Line impedance (in pu)	Line charging (in pu)
1-2	0.02+j0.06	j0.03
1-3	0.08+j0.24	j0.025
2-3	0.06+j0.18	j0.02
2-4	0.06+j0.18	j0.02
2-5	0.01+j0.03	j0.01
4-5	0.08+j0.24	j0.025

Table 2: Power Demand Data

Bus code	Real Power (Generation-Demand) in pu	Reactive Power (Generation-Demand) in pu
1	-	-
2	0.2	0.2
3	-0.45	-0.15
4	-0.4	-0.05
5	-0.6	-0.1

The number of input, hidden and output neurons is 12, 25 and 10 respectively. ANN is trained by means of Back Propagation algorithm and the MAPE obtained by means of this training is 1.964%.

To reduce this MAPE PSO trained ANN is used. The optimizing variables are the connecting weights of the ANN with the objective of minimization of error. Adaptive PSO is used with the population of 50 & with the inertia weight between 0.4 & 3. The program was executed on P IV 1 GHZ system for 50 runs. The MAPE obtained in this case in 0.785%. The results obtained by means of BPN trained and PSO trained ANN are compared against the conventional NR method and given in table 3 & 4.

Table 3: Bus Voltage Magnitudes in p.u

Bus code	NR method	BPN Trained ANN	PSO trained ANN
1	1.0600	1.0586	1.0578
2	1.0359	1.0276	1.0197
3	1.0076	0.9875	1.0025
4	1.0060	0.9837	1.0006
5	1.0002	0.9764	0.9909

Table 4: Voltage Angle in Radians

Bus code	NR method	BPN Trained ANN	PSO trained ANN
1	0	0.0002	0.0004
2	-0.0470	-0.0626	-0.0513
3	-0.0857	-0.1158	-0.0782
4	-0.0915	-0.1265	-0.1072
5	-0.1067	-0.1463	-0.1191

From the results obtained it is observed that the load flow solution using PSO trained ANN is compatible with standard NR method. The proposed algorithm eliminates the formation of Jacobin matrix and is applicable for larger power systems in which case NR fails to converge. Hence our proposed algorithm reduces the problem complexity to the greater extend.

VI. CONCLUSION

In this paper PSO trained neural network is used for solving the load flow problem. The algorithm can be used to solve the load flow problem under any load and system conditions. NR method fails to solve the load flow problem if the line resistance is much higher than line reactance. Also the proposed algorithm eliminates the formation of jacobian matrix and its inversion. For training purpose, we use the NR algorithm. Then for various loading conditions the outputs are directly obtained without forming jacobian and its inversion. This reduces the computational time. Training is done by powerful PSO algorithm which reduces the complexity of the problem. This algorithm utilizes the advantages of both PSO and neural network. Numerical results are shown for the system with balanced load and the algorithm gives accurate results with reduced complexity. The proposed work using PSO can be easily

ISSN 2278 - 8875
International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering
Vol. 1, Issue 6, December 2012

extended to large scale system. This work can be incorporated to find solution for power flow problem under contigent operation states. It works equally well for any ill conditioned problems also.

REFERENCES

1. Glem **W.** Stagg, Ahmed H. Alabiad methods in Power system analysis Macgraw-Hill, 1968.
2. AmgadA.EL_Dib,Hosam K.M.Youssef, M.M.EL-Metwally,Osrnan “Load Flow Solution Using Hybrid Particle Swarm Optimization” IEEE Trans.on Evalutionary Computation.
3. J.Arrillaga and N.R.Watson, “Computer modeling of electrical power systems” in Book, 2nd ed. Newyork: Wiley, 2001, pp.11-13.