IJCI Proceedings of Intl. XII. Turkish Symposium on Artificial Intelligence and Neural Networks, Vol.1, No.1, July 2003

# ELECTRODE CONTOUR OPTIMIZATION BY ARTIFICIAL NEURAL NETWORK WITH LEVENBERG-MARQUARDT ALGORITHM

Suna Bolat[1] Özcan Kalenderli[2]

[1, 2]Electrical & Electronics Faculty, Istanbul Technical University, Istanbul, Türkiye

[1]e-mail: bolats@itu.edu.tr [2]e-mail: ozcan@elk.itu.edu.tr

**Keywords**: Neural Network, Levenberg-Marquardt Algorithm, High Voltage, Optimization of Electrode Contour

## ABSTRACT

To obtain homogen electric field in any insulation is important for the reliability and life of electrical system. Otherwise, electric field is non-homogen and breakdown or partial discharge phenomena early become in the insulation. Therefore, electric fields on electrode contours and in insulation are calculated and optimized. In the determination of electric fields is used various methods as analytical, numerical or experimental methods. In this study, electrical fields in rod - plane electrode system are calculated by Charge Simulation Method and electrode optimization is performed by using Artificial Neural Network with Levenberg – Marquardt algorithm. With method used it has been obtained very fast desired results. It has been seen that method can be used efficiently for optimization of electrode contour in high voltage technology.

## 1. INTRODUCTION

The classical approach to design of an insulating system is based on the usage of simple geometrical formed elements. However, using spherical and cylindrical electrodes leads to nonuniform stress distribution and results increasing in the cost of insulation. For a better economy, it is necessary to have a uniformly distributed stress along the surface of insulator and electrode, and keeping the electric field as low as possible. Since the electric field heavily depends on geometric shape, a uniform field distribution in a high voltage arrangement cannot be obtained by such simple bodies as rings, cylinder or spheres. In order to have optimum contours with more complex geometries, it is necessary to optimize electrode and insulator contour by means of electrical field calculation.

Different methods have been developed for electrode and insulator contour optimization [1 - 3]. One of these methods, to obtain desired electric field distribution, electrode contours are modified iteratively by linear interpolation. In these iterative methods, since the electric fields have to be computed at every iteration step, computation time is very long. Therefore, iterative methods are not useful for every problem. Optimum stress distribution can be obtained by the method based on Artificial Neural Network (ANN) faster. The training set including a limited number of detailed field computations is sufficient for train the ANN.

During the last decade extensive research works have been carried out on the application of ANN in various fields. As a result, the literature on ANN is growing very rapidly [4 - 10]. ANN has been tried successfully on a very wide range of applications including machine vision, speech processing, sonar analysis, radar analysis, pattern recognition, robotic control etc. In electrical power systems, ANN has been used for accurate load forecasting, security evaluation, capacitor control, alarm processing etc. In high voltage systems, applications of ANN have been reported in the field of pattern recognition of partial discharges, pollution discharge modeling etc. Electrode and insulator contour optimization by ANN also have been presented [4 - 6].

In this study, electrode contour optimization in rod – plane electrode system to have a uniform field distribution along the rod electrode surface is performed by using artificial neural network. Among the various ANN structures presented so far, the multilayer feedforward network is used for supervised learning. To determine the connection weights of the ANN, the Levenberg – Marquardt algorithm is used in the training process.

## 2. ARTIFICIAL NEURAL NETWORKS

An ANN consists of a set of processing elements called neurons that interact by sending signal to one another along weighted connections. The connection weights, which can be determined adaptively, specify the precise knowledge representation. It is not possible to specify the weights beforehand, because the knowledge is distributed over the network. Therefore, a learning procedure is necessary in which the strengths of the connections are modified to achieve the desired form of activation function.

The learning procedure is divided into three types: supervised, reinforced and unsupervised. These three types of learning are defined by the types of error signal used to train the weights in the network. In supervised learning, an error scalar is provided for each output neuron by an external "teacher", while in reinforced learning the network is given only a global punish/reward signal. In unsupervised learning, no external error signal is provided, but instead internal errors are generated between the neurons, which are then used to modify weights.

In supervised learning the weights connecting neurons are set on the basis of detailed error information supplied to the network by an external teacher. In most cases the network is trained using a set of input-output pairs, which are examples of the mapping that the network is required to learn to compute. The learning process may therefore be viewed as fitting a function and its performance can thus be judged on whether the network can learn the desired function over the interval represented by the training set and to what extent the network can successfully generalize away from the points that it has been trained on.

### 2.1. MULTILAYER FEEDFORWARD NETWORK

The simplest network capable of supervised learning is a two-layer feedforward network consisting of an input layer and an output layer. Each neuron of the input layer receives a signal from all input neurons along connections with modifiable weights. But such two-layer feedforward networks can compute only linearly separable functions. However, it has also been shown that a feedforward network with more than one layer of adaptive weights can compute very complex functions.

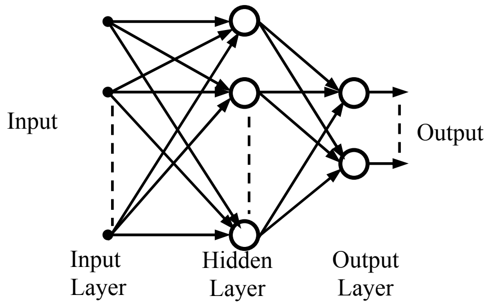


Figure 1 Schematic multilayer feed-forward ANN

The neurons in the network can be divided into three layers: input layer, output layer and hidden layers (Figure 1). It is important to note that in feedforward networks signals can only propagate from the input layer to the output layer via one or more hidden layers. It should also be noted that only the nodes in the hidden layers and the output layer, which perform activation function, are called ordinary neurons. Since the nodes in the input layer simply pass on the signals from the external source to the hidden layer, they are often not regarded as ordinary neurons.

### 2.2. LEVENBERG – MARQUARDT ALGORITHM

The back-propagation learning algorithm, which is a generalization of the Widrow-Hoff error correction rule, is the most popular method in training the neural network. The connection weights of the feedforward network are modified in the back propagation algorithm on the basis of the minimization of the performance function of network. By using different optimization technique for minimize the performance function, different learning algorithms are obtained. One of them is Levenberg – Marquardt algorithm. Levenberg – Marquardt algorithm is one of the high performance algorithms that can converge from ten to one hundred times faster than the back-propagation. It combines the best properties of back-propagation and Newton algorithms and eliminates their limits.
The basic step of the Levenberg – Marquardt algorithm is determination of Hessian matrix to determine the connection weights is used. Hessian matrix is second derivatives of the performance at the current values of weights and biases.

$$H(n)=\frac{\partial^2 E(n)}{\partial w^2(n-1)} \tag{1}$$

where H is the Hessian matrix, E performance function, w synaptic weights of network. Unfortunately, it is complex and expensive to compute the Hessian matrix for feedforward neural networks. Therefore, it is updated an approximate Hessian matrix at each iteration of the algorithm. The update is computed as a function of the gradient. When the performance function has the form of a sum of squares, as in this study, then the Hessian matrix can be approximated as follows:

$$H(n) = J^T(n)\,J(n) + \mu\, I \tag{2}$$

where $\mu$ is Marquardt parameter, I is identity matrix and J is Jacobian matrix that contains first derivatives of the network errors with respect to the weights and biases. The Jacobian matrix can be computed through a standard back-propagation technique that is much less complex than computing the Hessian matrix.

$$J(n) = \frac{\partial e(n)}{\partial w(n-1)} \tag{3}$$

In equation (3), e is a vector of network errors. The gradient can be computed as

$$g(n) = J^T(n)\,e(n) \tag{4}$$

and the weights can be updated by following equation,

$$w(n+1) = w(n) - [H(n)]^{-1}\, g(n) \tag{5}$$

When the scalar $\mu$ is zero, this is just Newton's method, using the approximate Hessian matrix. When $\mu$ is large, this becomes gradient descent with a small step size. Newton's method is faster and more accurate near an error minimum, so the aim is to shift towards Newton's method as quickly as possible. Thus, $\mu$ is decreased after each successful step (reduction in performance function) and is increased only when a tentative step would increase the performance function. In this way, the performance function will always be reduced at each iteration of the algorithm.

In general, Levenberg – Marquardt algorithm is not affected by slow convergence problem. The aim of the algorithm is finding the weight that makes the performance minimum [9 - 11].

### 2.3. NORMALIZATION OF INPUT – OUTPUT DATA

Since the input and output variables of the ANN have different ranges, the feeding of the original data to the network, leads to a convergence problem. It is obvious that the output of the ANN must fall within the interval of (0-1). In addition, input signals should be kept small in order to avoid a saturation effect of the sigmoidal function. So, the input-output patterns are normalized before training the network. Normalization by maximum value is done by dividing input-output variables to the maximum value of the input and output vector components. After the normalization, the input and output variables will be in the range of (0 to 1).

## 3. ELECTRODE CONTOUR OPTIMIZATION

The electrical field distribution on electrode surface should be uniform in order to minimize effects of partial discharge or breakdown phenomena and also get effective results from the electrode system. Nonuniform stress distribution might lead to breakdown in insulation material.

Electric fields along the surface depend on shape of the electrodes and insulators used in the high voltage arrangement. In order to get uniform stress distribution in a high voltage arrangement, electrodes and insulators contours should be optimized. Using the optimized electrode and insulator profiles not only obtains a uniform stress distribution but also keeps the electrical field value in acceptable limits. That is important for life of the electrode system and reliability of high voltage equipments [12, 13].

Rod – plane electrode system, which is very common in high voltage arrangement, has a nonuniform stress distribution. In this study, rod electrode contour is optimized by training an ANN and uniform stress distribution is obtained along the optimization region.

As the first step toward contour optimization by artificial neural network, the input – output patterns are to be provided for training. For this purpose electric field calculations are carried out by Charge Simulation Method [14]. For the training cases, end profile of the rod electrode is assumed to be circular (Figure 2). The electrode gap spacing with a potential difference of 10 kV is taken 2 cm.

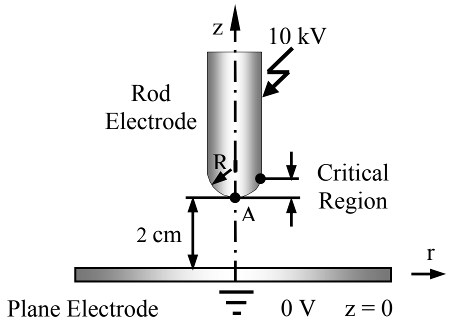


Figure 2 Rod – plane electrode system

For obtaining the different training patterns by means of electric field calculation, 10 different values of the radius of end profile of rod electrode are considered, e.g. 1 to 19 mm in steps of 2 mm. Hence, all together 10 sets of training patterns are obtained from the results of these field calculations. The electric fields are calculated at 7 different points on the circular surface of rod electrode for all cases, one is point A in the Figure 2, and 6 are in the critical region of circular surface of the rod electrode.

Table 1 Input Pattern Vectors for Rod – Plane Electrode System

| Electrode radius [mm] | Electric fields [kV/cm] | | | | | | |
|---|---|---|---|---|---|---|---|
| | Cylindrical coordinates of points (z, r) [mm] | | | | | | |
| | z = 20 r = 0 | z = 20.002 | z = 20.012 | z = 20.109 | z = 20.293 | z = 20.546 | z = 20.844 |
| 1 | 82.7200 | 73.7078 | 69.1567 | 67.4438 | 64.8832 | 61.6029 | 57.0954 |
| 3 | 18.4550 | 28.3780 | 30.5665 | 30.9938 | 30.7906 | 29.8017 | 28.8979 |
| 5 | 35.7250 | 56.6942 | 47.5656 | 24.7823 | 21.6146 | 20.5484 | 19.9858 |
| 7 | 16.5350 | 15.6948 | 14.8061 | 16.4203 | 16.2711 | 16.0771 | 15.8690 |
| 9 | 12.9110 | 12.7880 | 12.8228 | 13.8403 | 13.7754 | 13.5705 | 13.4164 |
| 11 | 11.5610 | 11.4856 | 11.3028 | 12.0945 | 12.1761 | 11.9911 | 11.8726 |
| 13 | 10.4610 | 10.4060 | 10.2476 | 11.0027 | 11.0197 | 10.8723 | 10.7774 |
| 15 | 8.4280 | 10.3354 | 8.9749 | 10.4879 | 10.1991 | 10.0694 | 9.9131 |
| 17 | 11.1850 | 10.2946 | 9.1885 | 8.9817 | 9.4331 | 9.4276 | 9.3151 |
| 19 | 9.8030 | 9.3311 | 8.7711 | 8.6099 | 8.9432 | 8.9268 | 8.8174 |

Coordinates of the point A are taken r = 0 and z = 20 mm for all cases. The other 6 points on the end profile are selected for all the contours considered at 6 fixed values of the z – coordinate, e.g. 20.002, 20.012, 20.109, 20.293, 20.546, 20.844 mm. The computed electric fields at the above mentioned 7 points are applied to the network as 10 sets of input pattern vectors, each with 7 items (Table 1).

The coordinates of the 6 points on the end profile as mentioned above are applied to the network as the output pattern vectors. Only the r – coordinates of the points are fed to the network, as the points on the end profile are selected at fixed values of z. All 10 output pattern vectors, each with 6 items, used for training are presented in Table 2.

Table 2 Output Pattern Vectors for Rod – Plane Electrode System

| Electrode radius [mm] | r – coordinates of points on the end profile [mm] | | | | | |
|---|---|---|---|---|---|---|
| | z – coordinates of points [mm] | | | | | |
| | z = 20.002 | z = 20.012 | z = 20.109 | z = 20.293 | z = 20.546 | z = 20.844 |
| 1 | 0.0654 | 0.1564 | 0.1540 | 0.7071 | 0.8910 | 0.9877 |
| 3 | 0.1570 | 0.1962 | 0.7775 | 1.0751 | 1.6339 | 2.1213 |
| 5 | 0.1581 | 0.3280 | 1.0800 | 1.6937 | 2.2700 | 2.8104 |
| 7 | 0.1583 | 0.4578 | 1.4010 | 2.0132 | 2.6061 | 3.1779 |
| 9 | 0.1585 | 0.4710 | 1.4079 | 2.3294 | 3.2253 | 3.8036 |
| 11 | 0.1587 | 0.4721 | 1.4180 | 2.3382 | 3.5482 | 4.1368 |
| 13 | 0.1588 | 0.4821 | 1.7020 | 2.6516 | 3.8675 | 4.4627 |
| 15 | 0.9810 | 0.4812 | 1.7251 | 2.9649 | 3.8823 | 5.0811 |
| 17 | 0.4650 | 0.7841 | 2.0310 | 2.9693 | 4.1973 | 5.4025 |
| 19 | 0.4712 | 0.7752 | 2.0381 | 3.2821 | 4.5118 | 5.7217 |

The network has 7 input neurons and 6 output neurons. With the input – output pattern vectors for training made available, the multilayer feedforward network is trained to give optimum end profile in such a way that a uniform stress distribution is obtained in the critical region of the end profile. After the training is completed, three different optimized end profiles are calculated by ANN for three desired uniform electric fields, e.g. 10, 12, and 24.4 kV/cm respectively, in the critical region [4 - 6].

## 4. APPLICATION OF ANN AND RESULTS

The ANN has 7 input neurons and 6 output neurons with single hidden layer. It has been seen that, taken the initial value for Marquardt parameter as $\mu_i$ = 0.001, increasing factor as $\mu^+$ = 10, and decreasing factor as $\mu^-$ = 0.1 give very good results for the network. The upper limit of the Marquardt parameter value is taken as $\mu_{max} = 10^{-10}$. All the above mentioned studies have been done with one hidden layer of 10 neurons after 1000 iterations. The number of neurons in the hidden layer is examined by varying the hidden layer neurons 7 to 22, which shows that 20 neurons in the hidden layer gives best results. In the hidden layer and the output layer, Sigmoid function is used as activation function.

After the training is completed, a test case, which is a pattern in the training set, is applied to the network and error is determined by comparing the actual output with output obtained from the artificial neural network. After network accuracy estimation, coordinates of rod electrode are determined by using the desired electric field values. All the above mentioned studies have been done by Matlab 6.5 Neural Network Toolbox.

It is to be noted here that in this study the error in training is represented by mean squared error (MSE) and the error in test is represented by mean absolute error (MAE).

$$\text{mse} = \frac{1}{N}\sum_{n=1}^{N}\frac{1}{2}\sum_{j\in C} e_j^2(n) \tag{6}$$

$$\text{mae} = \frac{1}{n_t\, n_o}\sum_{j=1}^{n_t}\sum_{k=1}^{n_o}\left[\frac{\left|d_{jk} - y_{jk}\right|}{d_{jk}}\right]100 \tag{7}$$

where $n_t$ is the number of test case, and $n_o$ is the number of output neurons, N denote the total number of patterns contained in the training set, $e_j(n)$ refers to the error signal, $d_{jk}(n)$ refers to the desired response and $y_{jk}(n)$ refers to the function signal at iteration n.

Training was continued, with the optimal network structure until the mean squared error met a fixed value of training accuracy, which is $10^{-13}$ (Figure 3). The training was stopped after 750 iterations (Table 3). r – coordinates of optimized end profiles as determined by applying the desired electric stresses are given in Table 4 for the network structure.

Table 1 An example for comparison of expected values and ANN outputs

| Number of neurons in the hidden layer = 20, Iteration number = 750 $\mu_i = 0.001$, $\mu^+ = 10$, $\mu^- = 0.,1$ | | | | | | |
|---|---|---|---|---|---|---|
| r – coordinates [mm] | | | | | | |
| Expected output | 0.1585 | 0.4710 | 1.4079 | 2.3294 | 3.2253 | 3.8036 |
| ANN output | 0.1585 | 0.4712 | 1.4077 | 2.3288 | 3.2249 | 3.8030 |

Table 2 Coordinates of the critical region

| Electric field values [kV/cm] | r – coordinates of the critical region [mm] | | | | | |
|---|---|---|---|---|---|---|
| | z – coordinates of the points [mm] | | | | | |
| | z = 20.002 | z = 20.012 | z = 20.109 | z = 20.293 | z = 20.546 | z = 20.844 |
| 10 | 0.0192 | 0.1389 | 1.1170 | 2.1817 | 3.3526 | 3.7427 |
| 12 | 0.0263 | 0.2542 | 0.4146 | 0.9929 | 2.2809 | 2.4718 |
| 24.4 | 0.0654 | 0.1564 | 1.1540 | 0.7071 | 0.8910 | 0.9877 |

The mean absolute error between expected output and computed values, as given by ANN output, is mae = $1.4759 \cdot 10^{-4}$ after 750 iterations for the example (Figure 3).

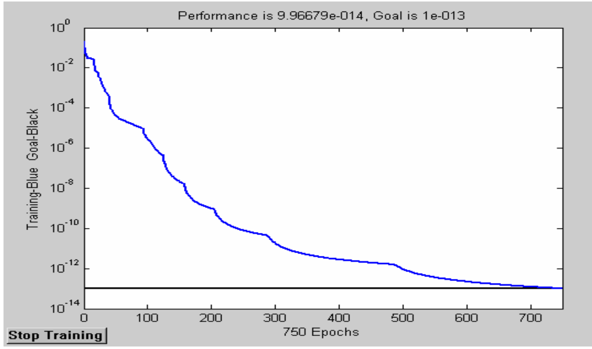


Figure 3 Change of performance function (mean squared error) with respect to iteration number

## 5. CONCLUSION

In this study, a multilayer feedforward network with Levenberg – Marquardt algorithm is designed for optimization of end profile of a rod – plane electrode system. It has been seen that, in the critical region, which is optimized, the mean absolute error is less than 0.1 for all cases. For optimization problem, the advantage of the use of ANN lies in the fact that the ANN is required to be trained only once. On the completion of training, the ANN gives the optimum results without further experiments or calculations which results in considerably saving in computation time. ANN also gives general results. By adding input-output patterns for training in order expand the range of expected outputs provides a result for a different problem.

Using Levenberg – Marquardt algorithm as learning algorithm is more efficient than the steepest descent and Gauss – Newton methods. Since the performance function will always be reduced at each iteration of the algorithm, it is possible to get minimum error with less iteration. This algorithm is much less complex than the algorithms which are using the other minimization techniques because it uses the approximate value of the Hessian matrix. Levenberg – Marquardt algorithm could be a good choice for lots of network structure by choosing available free parameters.

## REFERENCES

[1] Z. Stih, "High Voltage Insulating System Design by Application of Electrode and Insulator Contour Optimization", IEEE Trans. on Electrical Insulation Vol. EI-21, No. 4, pp. 579-584, 1986.

[2] H. Singer, "Computation of Optimized Electrode Geometries", Third International Symposium on High Voltage Engineering, paper 11-06, Milan, 1979.

[3] M. Abdel-Salam, E. K. Stanek, "Field Optimization of High Voltage Insulators", IEEE Transactions on Electrical Insulation Vol. 22, No. 1, pp. 47-56, 1987.

[4] P. K. Mukherjee, "Optimization of HV Electrode Systems by Neural Network using a New Learning Method", IEEE Transaction on Dielectric and Electrical Insulation, Vol. 3, No. 6, pp. 737-742, 1996.

[5] K. Bhattacharya, S. Chakravorti, P. K. Mukherjee, "Insulator Contour Optimization by a Neural Network", IEEE Trans. on Dielectrics and Electrical Insulation, Vol. 8, No. 2, pp. 157-161, 1994.

[6] S. Chakravorti, P. K. Mukherjee, "Application of Artificial Neural Networks for Optimization of

Electrode Contour", IEEE Transactions on Dielectric and Electrical Insulation, Vol. 1, No. 2, pp. 254-263, 1994.

[7] M. M. A. Salama, R. Bartnikas, "Determination of Neural Network Topology for Partial Discharge Pulse Pattern Recognition", IEEE Trans. on Neural Networks, Vol. 13, No. 2, pp. 446- 456, 2002.

[8] S. Bolat, Ö. Kalenderli, E. Önal, A. Merev, "Determination of impulse flashover voltage of HV insulators by artificial neural networks", ELECO'2002, Turkish Symposium on Electrical and Electronics Engineering, Bursa, pp. 154-158, 2002.

[9] M. Oguz, Determination of Electrical Strength of Insulating Materials by Artificial Neural Networks, MSc. Thesis, Institute of Science and Technology, Istanbul Technical University, 2001.

[10] S. Haykin, *Neural Networks: a Comprehensive Foundation***,** Upper Saddle River, N.J., Prentice Hall, 1999.

[11] S. H. Ngia, "Efficient Training of Neural Nets for Nonlinear Adaptive Filtering Using a Recursive Levenberg-Marquardt Algorithm". IEEE Trans. on Signal Process., Vol 48, pp. 1915-1927, 2000.

[12] T. Tkuma, T. Kawamoto, C. Honda, "Analysis of Electrical Field in Uniform-Field Electrodes", Ninth International Symposium on High Voltage Engineering, paper 8361-1, Graz, Austria, 1995.

[13] M. Özkaya, *High Voltage Technique*, Vol. I, Istanbul, 1996.

[14] H. Singer, H. Steinbigler, P. Weiss, "A Charge Simulation Method for the Calculation of High Voltage Fields", IEEE Transactions on Power Apparatus and Systems, Vol. 93, No.7, pp. 1660-1668, 1974.