Multi-Pattern-to-Single-Pattern Functional Approximations by combined Support Vector Machines and FeedForward Neural Networks

Vijaynarasimha H. Pakka
Research Student, Department of Electrical Engineering, Indian Institute of Science,
Bangalore 560 012, India
email: vijay@ee.iisc.ernet.in

***Abstract*— In many fields there are situations encountered, where a function has to be approximated to determine its output under new conditions. Some functions have one output corresponding to differing input patterns. Such types of functions are difficult to map using a function approximation technique such as that employed by the Multilayer Perceptron Networks. Hence to reduce this functional mapping to Single Pattern – to – Single Pattern type of condition, and then effectively estimate the function, we employ classification techniques such as the Support Vector Machines. This paper describes in detail such a combined technique, which shows excellent results for a practical application in the field of Power Distribution Systems.**

***Index Terms*— FeedForward Neural Networks, Support Vector Machines, Function Approximation, Pattern Classification.**

I. Introduction

FUNCTION approximation (FA) is typically the estimation of the output of an unknown function for a new input pattern, provided the function estimator is given sufficient training sets such that the unknown parameters defining the function are estimated through a learning strategy [1]. Function approximation is more commonly known as regression [2] in statistical theory. This function is usually a model of a practical system. The training sets are obtained usually by simulation of the system in real time. If the training set is given by,

$$\left\{\left(\bar{x}_1,\bar{y}_1\right),\left(\bar{x}_2,\bar{y}_2\right),\left(\bar{x}_3,\bar{y}_3\right),\ldots\ldots\ldots\left(\bar{x}_N,\bar{y}_N\right)\right\} \quad (1)$$

$\bar{x}$ = input pattern vector of size $n\times 1$, $\bar{y}$ = target vector of size $m\times 1$, N = number of patterns.

Then we need to estimate the functional relation between $\bar{x}$ and $\bar{y}$ i.e.,

$$\bar{y} = c_i f\left(\bar{x};\, t_i\right) \quad (2)$$

c_i = constants in the function, t_i = parameters of the function, $f: S \to \mathbb{R}$,

where $S \in \mathbb{R}^n$ is a closed bounded region, and if a and b are lower and upper limits of x, then $S = \left\{x \in \mathbb{R}^n \mid\ a_i \le x_i \le b_i, 1 \le i \le n\right\}$.

Function Approximation by Multilayer Perceptron Networks [3], [4] like the FeedForward Neural Networks (FFNNs) [5] is proven to be very efficient, considering various learning strategies like the simple Back Propagation or the robust Levenberg Marquardt [13] and Conjugate Gradient approaches. Radial Basis Function Networks (RBFNs) have also been applied to functional approximation [6]. Like networks with nonlinear transfer functions, RBFNs have the ability to represent arbitrary functions.

Assume there are m_0 number of variables in the function $f(\cdot) \Rightarrow\ f\left(\bar{x}_1,\bar{x}_2,\ldots\ldots\ldots,\bar{x}_{m_0}\right) \Rightarrow$ approximate function. Now if the true function is $F\left(\bar{x}_1,\bar{x}_2,\ldots\ldots\ldots,\bar{x}_{m_0}\right)$, then the FFNN equates this to

$$\sum_{i=1}^{m_1} \alpha_i\, \phi\left(\sum_{j=1}^{m_0} w_{ij}\,\bar{x}_{ij} + b_i\right) \quad (3)$$

Now the objective is to find the parameter values of m_1 and values of all w_{ij}'s, b_i's and α_i's, such that $\left|F(\cdot) - f(\cdot)\right| < \varepsilon$, for all $\bar{x}_1,\bar{x}_2,\ldots\ldots\ldots,\bar{x}_{m_0}$. In the next sections I shall consider the data sets obtained from a system as perfect measurements, i.e., no presence of noise. This allows us to investigate the entire problem without touching the topic of generalization, which is not of much debate in the idea behind this presentation.

In this paper, I shall demonstrate how, by effectively combining FeedForward Neural Networks and Support Vector

Classifiers, a complex Function Approximation problem can be broken down into a simple one and then solved to obtain accurate results. The proposed technique is applied to a real life problem, e.g., fault location in power distribution systems, and the results proving the efficiency of the technique.

II. Multi-Pattern-to-Single-Pattern Functions

Let us look at the problem of FA as a mapping problem, where, by one-to-one mapping we mean that each input vector has a corresponding and unique target vector. These mappings are simple to model by FFNNs. This relates to a function that has one output for each input. But, this is not the case in many fields. For example, let us study the case of a sine function. Assume that a system has the characteristic shown in Fig. 1a, which has to be estimated. Let the estimation be done by a FFNN with nonlinear transfer functions. The data sets that are available for training of the FFNN are the data corresponding to the two cycles a and b (red and blue). Let the data points be obtained on the sine curve at the rate of s samples per cycle.

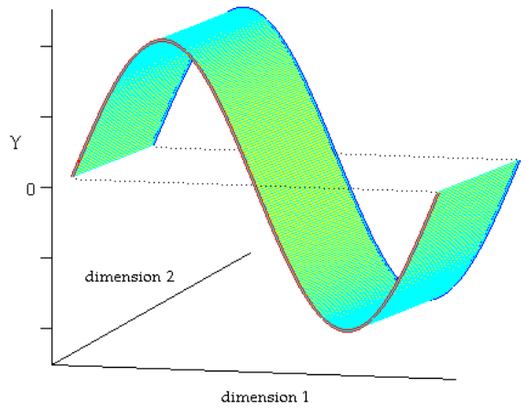

Fig. 1a. Sine wave characteristics of a sample system. Red sine represents cycle 'a' and blue sine represents cycle 'b'.

For convenience Fig. 1a is redrawn as Fig. 1b. Inputs X_{ai} and X_{bi} have same output Y_i. This value is *stored* by the FFNN in the form of a straight line. For both the inputs running through one cycle, we have a set of such straight lines with varying amplitudes (Fig. 2).

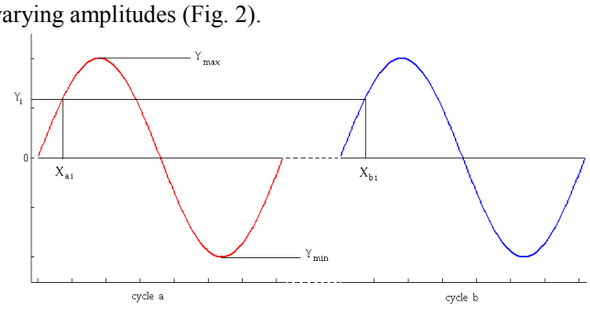

Fig. 1b. Simpler representation of Fig. 1a.

Now suppose there exists an intermediate sine cycle between cycles a and b. if p has similar shape and size as of a and b, then, we can estimate its Y throughout the cycle just by noting the Y values at corresponding intermediate point X_p in Fig. 2. This estimation of Y turns out to be equal to that at X_a or X_b.

Now instead of p being similar to a or b, suppose it to be of different size as shown in Fig. 3. Now, as in Fig. 2, if we estimate Y's at $X = X_p$, the results would not match with that of the true function represented by p. this is due to the fact that the curves joining the two sets of vertical points in Fig. 2 are still straight lines, though in reality they are of the shape of curves with amplitudes (Y's) at X_p different from that at X_a or X_b.

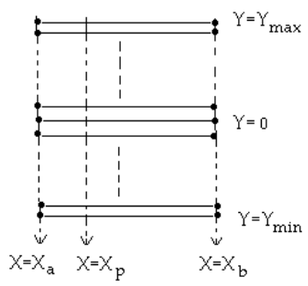

Fig. 2. Graphical depiction of "how FFNN stores input-to-output functional relationship".

Let us name this type of mapping as *Two-way mapping* or *Multi-Pattern-to-Single-Pattern mapping* in general, because, estimation of the actual function is the first FA problem, and estimation of the shapes of the curves (lines in Fig. 2) is the second FA problem, i.e., two different input patterns X_{ai} and X_{bi} correspond to a single output pattern Y_i. This is illustrated graphically in Fig. 2 by straight lines.

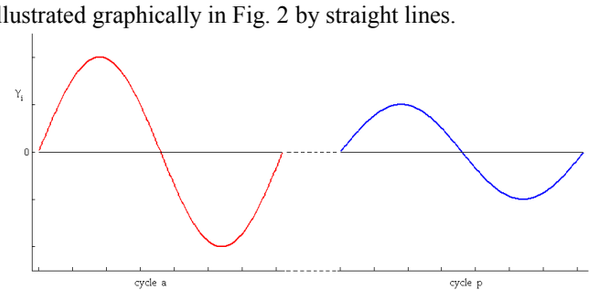

Fig. 3. Different characteristics of the same system.

The misestimating of p is mainly due to insufficient data (cycles) between a and b. Even if there were data between a and b, this would have called for a strain on the FFNN to learn the entire input space. This is because it has to learn in both directions, one in the direction of the sine propagation and the other in the direction of the vectors joining a and b. To relieve the FFNN of this burden [7], [8], the datasets are labeled and correspondingly classified using Support Vector Classifiers

(SVCs), which are then combined suitably with the FFNNs so as to give an effective approximation to the overall true function of the system under study. The FFNNs are trained using the Levenberg Marquardt algorithm [13] with regularization term, and the SVCs are trained by the simple but yet very fast algorithm of Sequential Minimal Optimization (SMO) [15].

III. Function Approximation by Combined FFNNs and SVCs

We have described in detail, what I mean by the term Multi-Pattern-to-Single-Pattern Functional Mappings. These types of characteristics are often encountered in the modeling of practical systems. Hence their detailed analysis is of good relevance. Also, from the perspective of pure mathematics, this problem possesses quite importance for detailed analysis to be done. To describe and apply the proposed approach to a practical system, we shall consider a live topic in the field of power engineering. Fault Location in Transmission and Distribution Industry has received quite interest in the last two decades. Ever since Neural Nets have risen as powerful Function Approximators, the area of fault location has received much more attention [9]. We shall briefly describe here, the problem of Fault Location in Distribution Systems.

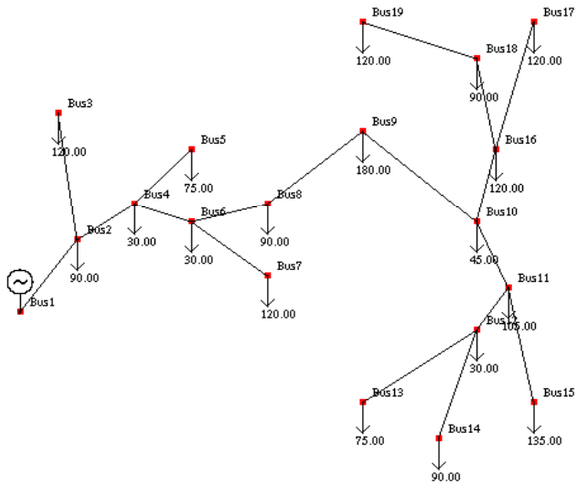

Fig. 4. A practical 19-Bus 11KV distribution system feeder.

Consider a practical 11 KV, 19 bus Distribution Feeder shown in Fig. 4. Each bus is a distribution transformer with a specified load. The feeder line has a resistance (R) and reactance (X) of 0.0086 *and* 0.0037 $p.u./km$ respectively. As R/X ratio is fixed, let us consider X as the only variable.

For the detection fault location, we need to consider various practical aspects involved in the day-to-day operation of a distribution system. In a single day we have various loading patterns, which have to be simulated, and also we need to consider various types of faults that occur in a realistic scenario. During fault conditions, if we consider the three-phase voltage and current measurements at the substation (bus 1) as our input elements, we can predict the location of fault by the output of the function estimator. This output is the reactance of the line, which in turn is the length of the faulty part of the line measured from bus 1. This is a *single-pattern-to-single-pattern* type of functional mapping, as each measurement vector produces a corresponding and unique output pattern. The other practical factors mentioned above, lead to the *Multi-Pattern-to-Single-Pattern Functional Mapping*, which has to be mapped exactly for the estimation of fault location in real time. For generating the data sets for training, the following procedure is adopted:

- A Short Circuit is simulated with a particular type of fault (Line-Ground (LG), Line-Line (LL), Line-Line-Ground (LLG), Symmetrical 3phase) at a particular bus, and at a particular Source Short Circuit (SSC) level (this is to simulate the loading patterns of the system).
- Measurements are noted at the substation.
- The 6×1 input pattern (three voltages and three currents) is reduced to 3×1 using Principal Component Analysis (PCA) (useful in viewing the dataset).
- Now the SSC level, fault type, and the fault buses are varied throughout their range, individually, and the data set is built up. The SSC range is from 20MVA to 50 MVA in steps of 5MVA.

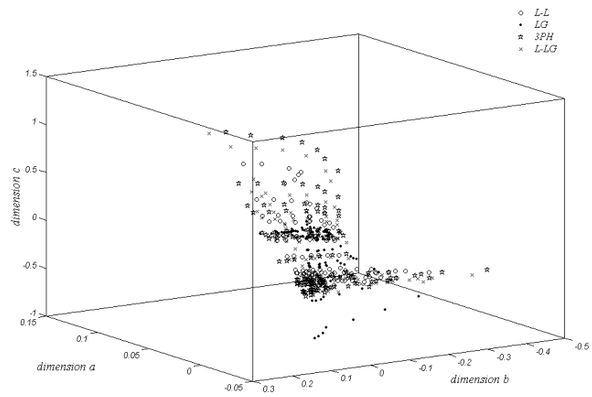

Fig. 5. Dataset of the complete function estimation problem.

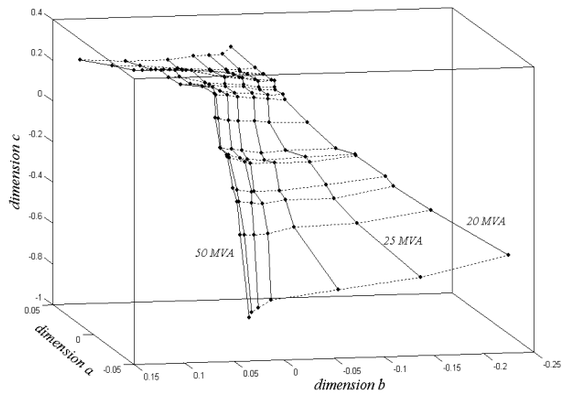

Fig. 6. Dataset corresponding to LG fault. Each dot represents fault on a bus, the solid curves represent variation of fault position (buses), and the dotted curves represent variation of the SSC level.

We see from Fig. 5 that estimating this complex function is quite difficult for an individual FFNN with any architecture. Hence, the first Function Breakup is by labeling the data according to their fault types and then classifying them by a SVC. In real time, this SVC block classifies the type of fault of an input pattern and the function estimator corresponding to this fault type does the remaining job. This is seen from Fig. 6, where the function looks less complex and can be modeled with less difficulty.

A SVC is trained, with data of each solid curve being labeled according to their SSC levels. As there are 7 SSC levels that are simulated, we have 6 binary classifiers, which classify patterns to a particular SSC level for further use by the function approximators. Now the work of the function approximators (in this case the FFNN) is cut down to estimation of the solid curves, i.e., data relating to one fault type and one SSC level. If n is the number of SSC levels that are simulated, then the number of classifiers chosen is (n – 1). Thus, there are 6 binary classification problems for each of the SSC level classifiers 'SVM a' to 'SVM d' to solve during training process, e.g., Classifier 1 classifies faults of 20 MVA and 25 MVA. 'SVM a' in Fig. 7 refers to SSC level classifier that is trained with Line to Ground faults. The function value $f(x)$ of (A6) points to the class the pattern belongs to i.e., each SVC outputs the pattern as a positive or negative function value, which is indicative of it belonging to either class.

TABLE I
CLASSIFYING LG FAULTS OF TWO LEVELS

Classifier No	Classes (MVA)	37MVA	38 MVA
1	20 – 25	-4.3428	-4.2143
2	25 – 30	-3.5312	-3.1255
3	30 – 35	**-1.9712**	-2.5783
4	35 – 40	**1.0923**	**0.2345**
5	40 – 45	3.0093	**1.7389**
6	45 – 50	7.2876	3.8578

Table I describes the classification of 37 MVA and 38 MVA SSC level faults as that of 35 MVA and 40 MVA SSC levels respectively. The results correspond to LG faults of levels 37 MVA and 38 MVA, simulated on a bus. The $f(x)$ value of the 37 MVA fault changes sign at classifier nos. 3, 4 (in third column of Table I - the value of $f(x)$ changes from -1.9712 to +1.0923) and the common class between these two classifiers being 35 MVA, we classify this fault as one that occurred in the group of 35 MVA. Similarly for the 38 MVA fault, which is categorized as belonging to 40 MVA class. The proposed combined approach to the function estimation problem relevant to this application is depicted in the form of a block diagram below. Also, the results in Table II show that the estimation of the unknown function (B7) by the proposed approach has errors in the negligible range of 0.5 – 1.0 %.

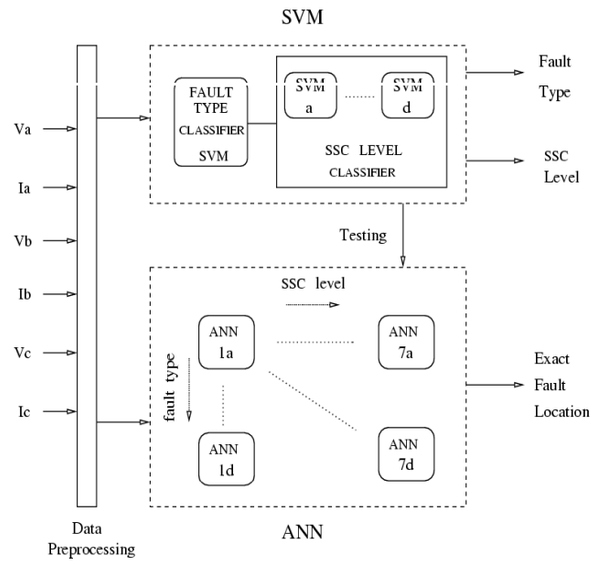

Fig. 7. Block Description of the proposed approach.

TABLE II
TRAINING PATTERNS AND TARGETS FOR LG FAULTS AT A SOURCE SHORT CIRCUIT LEVEL OF 35 MVA AND THE CORRESPONDING OUTPUTS.

Bus	V_a	V_b	V_c	I_a	I_b	I_c	Target	Output
Source Short Circuit Level: 20 MVA								
2	0.66	0.92	1.00	23.5	1.91	2.06	0.1110	0.1112
3	0.88	0.96	1.01	11.4	1.99	2.10	0.2960	0.2934
4	0.77	0.94	1.01	17.5	1.95	2.13	0.1665	0.1672
5	0.84	0.95	1.01	13.9	1.98	2.12	0.2220	0.2221
6	0.82	0.95	1.01	14.9	1.98	2.14	0.2035	0.2047
7	0.87	0.96	1.01	11.6	2.00	2.13	0.2775	0.2772
8	0.88	0.96	1.01	10.9	2.02	2.15	0.2960	0.2929
9	0.92	0.97	1.01	8.34	2.06	2.15	0.4070	0.4084
10	0.95	0.98	1.01	6.12	2.08	2.15	0.5920	0.5915
11	0.96	0.98	1.01	5.70	2.08	2.14	0.6475	0.6471
12	0.96	0.98	1.01	5.46	2.08	2.14	0.6845	0.6822
13	0.97	0.99	1.01	4.83	2.08	2.13	0.8140	0.8156
14	0.97	0.99	1.01	4.76	2.08	2.13	0.8325	0.8334
15	0.97	0.99	1.01	4.75	2.08	2.13	0.8325	0.8351
16	0.96	0.98	1.01	5.70	2.08	2.14	0.6475	0.6432
17	0.97	0.99	1.00	4.61	2.08	2.12	0.8695	0.8677
18	0.97	0.99	1.01	4.75	2.08	2.13	0.8325	0.8339
19	0.97	0.99	1.00	4.27	2.08	2.12	0.9805	0.9818

IV. CONCLUSION

In this paper it is proved that a complex function approximation problem can be solved efficiently by simple neural networks such as FeedForward Neural Networks and Support Vector Classifiers, but only through the techniques such as the one proposed here, which combines both of the above neural networks to reduce the complex function to a simpler one. The idea behind this type of function reduction is substantiated with results from application of the proposed

technique to a real life problem. The problem chosen for the case study was the fault location problem in power distribution systems, as it has some unique features, which result in a complex function approximation problem.

APPENDIX

A. Support Vector Classifiers

For the training of the SVM the Sequential Minimal Optimization (SMO) algorithm is used, which is very fast and robust for the present problem.

For training data from the i^{th} and j^{th} classes, we solve the following binary classification problem:

$$\begin{aligned} \min_{w^{ij},b^{ij},\xi^{ij}} \quad & \frac{1}{2}(w^{ij})^T w^{ij} + C\sum_t \xi_t^{ij} \\ & (w^{ij})^T \phi(x_t) + b^{ij} \geq 1-\xi_t^{ij}, \quad if\ y_t = i, \\ & (w^{ij})^T \phi(x_t) + b^{ij} \leq -1+\xi_t^{ij}, \quad if\ y_t = j, \\ i.e., \quad & y_t[(w^{ij})^T \phi(x_t) + b^{ij}] \geq 1-\xi_t^{ij}, \quad t = 1,\ldots,k, \\ & \xi_t^{ij} \geq 0. \end{aligned} \tag{A1}$$

where $\phi : \mathbb{R}^n \to H$ is some nonlinear function.

The training data x_i are mapped to a higher dimensional space by the function ϕ and C is the penalty parameter. The training data x_i are mapped to a higher dimensional space by the function ϕ and C is the penalty parameter. The term $\frac{1}{2}(w^i)^T w^i$ is minimized because this would mean maximizing the margin $2/\|w_i\|$ between the two classes of training data. The penalty term $C\sum_t \xi_t^{ij}$ is required when the classes are not linearly and perfectly separable. The Lagrangian for the above problem is:

$$\begin{aligned} L(w,b,\xi_t;\alpha,\beta) &= \frac{1}{2}(w)^T w + C\sum_{t=1}^{N}\xi_t \\ &+ \sum_{t=1}^{N}\alpha_t[1-\xi_t - y_t(w^T\phi(x_t)+b)] - \sum_{t=1}^{N}\beta_t\xi_t \end{aligned} \tag{A2}$$

By the Wolfe dual problem [16] of maximizing this lagrangian, we get the optimum weights and bias terms as:

$$w = \sum_{t=1}^{N}\alpha_t y_t \phi(x_t) \quad \sum_{t=1}^{N}\alpha_t y_t = 0 \quad C = \alpha_t + \beta_t \tag{A3}$$

$$1 \leq t \leq N,\ \alpha \geq 0, \beta \geq 0,\ 0 \leq \alpha_t \leq C$$

Substituting these in the Lagrangian, we get the dual problem to be solved to get the optimum Lagrangian variables:

$$\begin{aligned} &\max\ L = q(\alpha) = \sum_{i=1}^{N}\alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{i=1}^{N} y_i y_j \phi^T(x_i)\phi(x_j)\alpha_i\alpha_j \\ &subject\ to \quad \sum_{i=1}^{N} y_i\alpha_i = 0, \quad 0 \leq \alpha_i \leq C \quad 0 \leq i \leq N \end{aligned} \tag{A4}$$

Instead of the dot product $\phi^T(x_i).\phi(x_j)$ we use the dot product Kernel $K(x_i,x_j) = e^{\frac{-[(x_i-x_j)^T(x_i-x_j)]}{2\sigma^2}}$ to avoid the explicit calculation of the function ϕ.

The SMO algorithm searches through the feasible region of the dual problem and maximizes the above objective function, find finding the optimal values of the Lagrangian multipliers. The conditions for optimality can be stated as

$$\begin{aligned} &\alpha_i = 0 \ \Rightarrow\ y_i f(x_i) \geq 1 \\ &0 < \alpha_i < C \ \Rightarrow\ y_i f(x_i) = 1 \\ &\alpha_i = C \ \Rightarrow\ y_i f(x_i) \leq 1 \end{aligned} \tag{A5}$$

Where $f(x)$ is given by,

$$f(x) = \sum_{i\in S}\alpha_i y_i K(x_i,x) - b \tag{A6}$$

Here $S = \{i : \alpha_i > 0\}$ & $b = y_j - \sum_{i\in S}\alpha_i y_i K(x_i,x_j)$ for some j such that $0 < \alpha_j < C$.

B. Functions in the Fault Location Problem

Fig. 8 shows a general fault occurring on the distribution system feeder. The substation is considered as the source or the sending end of the power, while the load is the receiving end. During steady state fault conditions with fault resistance to ground, the source supplies most of the short circuit MVA to the fault point.

There is a definite relation between the measurements at the sending end, and the distance of fault from this end, for the faulted phase. Consider a fault of a general type occurring on a bus (or on a line section) at a distance of p from the Source side.

$V_{i(0)}^{abc}, V_{i(f)}^{abc}$: Voltages at bus i, initially and during fault conditions.

Z_{ii}^{abc} : Driving Point Impedance of bus i.

I_f^{abc}, Z_f^{abc} : Fault current, fault impedance

$V_{p(f)}^{abc}$: During fault voltage at fault point p.

$$V_{p(f)}^{abc} = Z_f^{abc} I_f^{abc} \tag{B1}$$

Voltage of bus i during fault is

$$V_{i(f)}^{abc} = V_{i(0)}^{abc} - Z_{ii}^{abc} I_f^{abc} \tag{B2}$$

Similarly voltage at fault point p during fault is

$$V_{p(f)}^{abc} = V_{p(0)}^{abc} - Z_{pp}^{abc} I_f^{abc} \tag{B3}$$

Substituting (B1) in (B3),

$$Z_f^{abc} I_f^{abc} = V_{p(0)}^{abc} - Z_{pp}^{abc} I_f^{abc} \tag{B4}$$

The fault current is obtained as

$$I_f^{abc} = \left(Z_f^{abc} + Z_{pp}^{abc}\right)^{-1} .\ V_{p(0)}^{abc} \tag{B5}$$

Substituting the value of I_f^{abc} from (B5) in (B2),

$$V_{i(f)}^{abc} = V_{i(0)}^{abc} - Z_{ip}^{abc} \,.\, (Z_f^{abc} + Z_{pp}^{abc})^{-1} \,.\, V_{p(0)}^{abc} \qquad \text{(B6)}$$

This is the value of voltage at bus i, during steady state fault conditions. From (B6), it is seen that, the voltage at bus i during fault is a function of initial voltage at buses i and p, driving point impedance of bus p, transfer impedance between buses i and p, and the fault impedance. By considering $V_{i(f)}^{abc}$ as a measurement at the sending end, we obtain the distance of the faulty bus p from the sending end. (The distance is implicit of the terms Z_{ip}^{abc} and Z_{pp}^{abc}).

In simple analogy, (B6) is $\overline{y} = f\left(\overline{x}\right)$ (B7)

where,

$\overline{x} = \left(V_s^{abc}, I_s^{abc}, R_f^{abc}\right) =$ Measurements at sending end,

$\overline{y} = g(Z_{pp}^{abc}, Z_{ip}^{abc}) =$ Distance of fault from sending end.

g is a function relating the distance of fault bus from the source, and the corresponding terms of the Z-bus matrix.

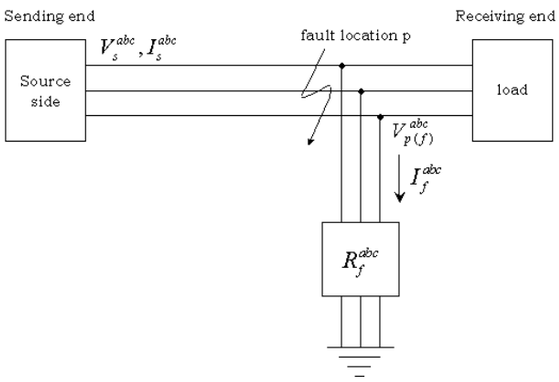

Fig. 8. 3-phase Radial network with general fault at distance p from the source side.

The relation in (B7) will become complex, once varying load conditions, fault resistance, and different types of faults on a feeder, are incorporated. This function would then correspond to numerous inputs mapping onto a single target, e.g., a LG fault occurring at the same location on a feeder, but at different SSC levels. Such type of mapping is difficult to model by a single FFNN. Hence, this paper proposed a new, combined approach, wherein the SVM breaks the complexity of (B7), and the FFNN estimates the unknown parameters in (B7) based on supervised learning.

REFERENCES

[1] D. A. Sprecher, "On the structure of continuous functions of several variables," *Transactions of mathematical Society*, 115, 1964, pp. 340–355.

[2] N. R. Draper, and H. Smith, *Applied Regression Analysis*, 2nd Ed., 1981, NewYork: John Wiley & Sons.

[3] K. Hornik, "Multilayer feedforward networks are universal approximators," *Neural Networks*, 2, 1986, pp. 359–366.

[4] E. K. Blum, and L. K. Li, "Approximation theory and feedforward networks," *Neural Networks*, Vol. 4, 1991, pp. 511–515.

[5] K. S. Narendra and K. Parthasarathy, "Identification and control of dynamic systems using neural networks," *IEEE Trans. Neural Networks*, Vol. 1, pp. 4-27, 1990.

[6] N. Dyn, "Interpolation and Approximation by radial and related functions," *C.K. Chui, L.L. Schumaker and J.D. Ward, Eds., Approximation Theory* VI: Vol. 1 (Academic Press, 1989) pp. 639–659.

[7] B. Irie, and S. Miyake, "Capacity of Three Layered Perceptrons," *Proc. IEEE ICNN 1*, 1988, pp 641–648.

[8] K. Hornik, "Approximation capabilities of multilayer feedforward networks," *Neural Networks*, Vol. 4, 1991, pp. 251–257.

[9] T. S. Bi, Y. X. Ni, C.M. Shen, F. F. Wu, and Q. X. Yang, "A novel radial basis function neural network for fault section estimation in transmission network," *Proc 5th Int Conf. Adv in Power System Ctrl, Oprn and Mgmt,* Hong Kong, 2000, Volume: 1, pp. 259 - 263.

[10] T. Masters, *Advanced algorithms for neural networks: a C++ sourcebook,* 1995, New York: Wiley.

[11] Chih-Wei Hsu, and Chih-Jen Lin, "A comparison of Methods for Multi-Class Support Vector Machines," Available at: www.csie.ntu.edu.tw/~cjlin/papers

[12] Klaus-Robert Muller, Sebastian Mika, Gunnar Ratsch, Koji Tsuda, and Bernard Schölkopf, "An Introduction to Kernel-Based Learning Algorithms," *IEEE Trans. Neural Networks*, Vol. 12, No. 2, March 2001, pp. 181–201.

[13] M. T. Hagan, and M. Menhaj, "Training feedforward networks with the Marquardt algorithm," *IEEE Trans. Neural Networks*, vol. 5, pp. 989-993, Nov. 1994.

[14] C. J. C. Burges, "A tutorial on support vector machines for pattern recognition," *Data Mining and Knowledge Discovery*, vol. 2, no. 2, 1998.

[15] J. C. Platt, "Fast training of support vector machines using sequential minimal optimization," *Adv in Kernel Methods: Support Vector Machines*, B. Schölkopf, C. Burges, and A. Smola, Eds. Cambridge, MA: MIT Press, Dec. 1998.

[16] R.Fletcher, *Practical Methods of Optimization*, Wiley, New York, 2000.

[17] J. C. Mishra (Ed), *Computing and Information Sciences: Recent Trends.* Narosa Publishing House, New Delhi, 2003.