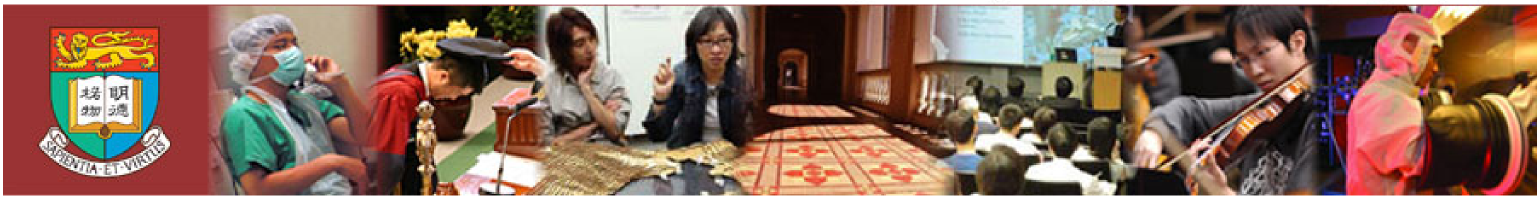

Title	Advanced Fault Section Estimation System for Power Networks Based on Hybrid Fuzzy System and Radial Basis Function Neural Network
Author(s)	Bi, T; Ni, Y; Wu, FF
Citation	The 33rd Annual North American Power Symposium, College Station, USA, October 15-16, 2001, p. 99-104
Issued Date	2001
URL	http://hdl.handle.net/10722/57286
Rights	©2001 IEEE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE.

Proceedings of NAPS 2001, Texas A&M University, College Station, USA, October 15-16, 2001

Advanced Fault Section Estimation System for Power Networks Based on Hybrid Fuzzy System and Radial Basis Function Neural Network

Tianshu Bi, St. M, IEEE Yixin Ni, S. M, IEEE Felix F. Wu, Fellow IEEE

Center for Electrical Energy System
Dept. of Electrical & Electronic Engineering
The University of Hong Kong, Hong Kong SAR

Abstract Although the radial basis function neural network (RBF NN) offers a potential solution for fault section estimation (FSE) in power networks, it has to be totally retrained for the case of power network topology change or power network expansion and cannot provide any explanations for its diagnosis results due to the black-box nature of the neural network. In this paper, the functional equivalence between RBF NN and fuzzy system (FS) is built up for FSE problem throughout the neural network training process. Furthermore, based on this point, a novel retraining strategy is presented for RBF NN, which can extract the unchanged knowledge from the original RBF NN and then insert the knowledge back to the new RBF NN about the changing part of the power network in the case of network topology change or expansion. The retraining strategy has been implemented and tested in a 4-bus power system. The simulation results show that the advanced FSE system with hybrid FS and RBF NN works successfully and efficiently in power networks.

Keywords fault section estimation, fuzzy system, radial basis function neural network, retraining strategy, power networks

I. INTRODUCTION

To reduce power supply interruption and enhance service reliability, fault section estimation (FSE) should be implemented quickly and accurately in order to isolate the faulted elements from the rest of the system and to take proper countermeasures to recover normal power supply. However, FSE is difficult because of the large amount of information to be dealt with and the FSE speed and accuracy required, especially for the cases with malfunctions of relays and circuit breakers, or multiple faults at the same time.

Radial basis function neural network (RBF NN) provides a potential solution for FSE in power networks [1-2] because of its universal approximation capability [3], automatically determined structure in training process and faster training convergence and better generalization capability as compared with other approaches. However RBF NN has no adaptation capability for the case of power network topology change or power network expansion, i.e., the corresponding FSE system has to be totally retrained, which is extremely time-consuming. Power network topology change or power network expansion is inevitable in power systems due to the maintenance, restoration or power system development. Since neural networks imply the knowledge in the network architecture and all the numeral weights, it is very difficult to extract the explicit rules from the structure and numerals, which is the bottleneck of the adaptation capability of neural networks.

Considerable efforts [4-7] have been made on interpreting the intrinsic knowledge of neural networks. [4] attempts to train the BP NN from hints, which allows for general information to be used instead of just input-output training samples, while [5-7] aim at interpreting the RBF network. It can be seen that the RBF network, a locally tuned neural network, is much easier to be understood than the BP NN. In [5], the normalized RBF NN is represented by probabilistic rules, however probabilistic rules are not so easy to be obtained in real applications due to the required large number of the historical data. [6-7] put their emphasis on building up the functional equivalence between the RBF NN and the fuzzy system (FS), which in many cases is highly desirable since FS is complementary with RBF NN and can be easily understood and used by domain experts. Though the initial results are given, there still exist some imperfections. The conclusion part of the fuzzy rule is limited to be a constant instead of general fuzzy proposition. The implication of the fuzzy rule is ignored. The fuzzifier and the defuzzifier have not been considered either.

In this paper, the functional equivalence between RBF NN and FS in the feed-forward calculation is derived in a more general form as compared with that mentioned in the reference. Besides, it is pinpointed that for FSE problem this functional equivalence holds throughout the training process of the RBF NN. Furthermore, based on the functional equivalence, in the case of power network topology change or power network expansion a novel retraining strategy is presented for RBF NN, which can extract the unchanged knowledge from the original RBF NN and then insert the knowledge back to the new RBF NN about the changing part of the power network and thus enhances its adaptability, reduces its retraining time and improves its FSE capability.

The paper is organized as follows. Section 2 briefly introduces the RBF NN and the FS used in FSE. The functional equivalence between RBF NN and FS is analyzed in Section 3. Section 4 proposes the novel retraining strategy for the case of power network topology change or power network expansion. Section 5 shows the computer simulation results and in section 6 we present conclusions.

II. RBF NN & FS USED IN FSE

A. RBF NN for FSE

The RBF NN [8-9] consists of feed-forward architecture with an input layer, a hidden layer of RBF units and an output layer of linear units (Fig. 1). The single output network is used as the illustrative example. All the obtained results can be extended to the multi-output network with ease.

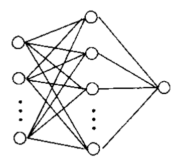

Fig. 1 The structure of RBF network

The input layer simply transfers the input vector to the hidden neurons, which form a localized response to the input pattern. Typically the activation function of the hidden neuron is chosen as a Gaussian function:

$$\varphi_i(x)=\exp\left(-\frac{\|x-c_i\|^2}{\sigma_i^2}\right)\quad (i=1,\ldots,n_h) \tag{1}$$

where $\varphi_i(x)$ is the output of the i^{th} hidden neuron; n_h is the number of the hidden neurons; x is the input vector; c_i and σ_i are the center (or weight) and the spread of the corresponding Gaussian function respectively.

The output layer generates the desired output through linear mapping of the outputs of the RBF layer. The output of the RBF NN will accordingly be:

$$o=\sum_{i=1}^{n_h} w_i\cdot\varphi_i(x) \tag{2}$$

where w_i is the connection weight from the i^{th} hidden neuron to the output neuron.

Several learning algorithms [8, 10-11] have been proposed to identify the parameters (c_i, σ_i and w_i). Due to the localized architecture of RBF NN, the learning algorithms usually can determine the structure automatically [10-11] in the training process and have faster training convergence than the learning algorithms of other neural networks.

For FSE in power networks, the relays and circuit breakers states (0 or 1) are taken as the inputs to the RBF NN, while the states (faulted or normal) of system elements, such as transmission lines, transformers and buses, as the outputs. Some typically fault scenarios are selected to make up the training sample set.

It should be noted that the RBF NN for FSE has two significant characteristics for interpreting the trained neural networks. One is that the inputs of the network are binary variables, which makes the centers of the RBF neurons are binary variables too. The other is for FSE the spreads of all RBF neurons are same [1], i.e., $\sigma_1=\ldots=\sigma_i=\ldots=\sigma_{n_h}=\sigma$.

B. FS for FSE

The basic configuration of FS for FSE is shown in Fig. 2.

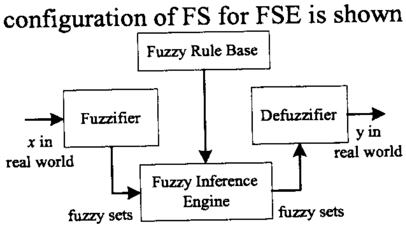

Fig. 2 FS for FSE

Fuzzy rule base is the basis of the inference. In general, fuzzy rule can be expressed as:

IF x_1 is c_{i1} and …and x_n is c_{in}, THEN y is w_i (3)

where $x=[x_1, x_2, \ldots, x_n]^T$ and y are the input and output linguistic variables respectively, while $c_{i1}, \ldots, c_{in}$ and w_i are the corresponding linguistic value (fuzzy sets). Let M be the total number of rules in the fuzzy rule base, then i=1, 2, …M.

For FSE in power networks, the input linguistic variable x is composed of all the relays and circuit breakers, while $c_{i1}, \ldots, c_{in}$ are the fuzzy sets defined on the corresponding status of these protective devices, i.e. operated or non-operated. Likewise, y is the concerned system element, while w_i is the corresponding possibility that this element is in fault.

In a fuzzy inference engine, fuzzy logic principles are used to combine the fuzzy rules in the fuzzy rule base into a mapping from input fuzzy set A' to output fuzzy set B'. Suppose μ represents membership function, the output of the inference engine will be:

$$\mu_{B'}(y)=\sup_x t\left[\mu_{A'}(x),\mu_{x\to y}(x,y)\right] \tag{4}$$

where sup and t represent support and t-norm operator respectively. $\mu_{x\to y}(x, y)$ is the membership function of the IF-THEN rule, which can be interpreted by different methods [12]. If the Mamdani implication [12], the most widely used implications in FS, is selected, $\mu_{x\to y}(x, y)$ can be calculated by:

$$\mu_{x\to y}(x,y)=t\left[\mu_{c_{i1}}(x_1),\mu_{c_{i2}}(x_2),\ldots,\mu_{c_{in}}(x_n)\right]\cdot\mu_{w_i}(y) \tag{5}$$

However, both input A' and output B' are fuzzy sets instead of crisp values in real world. Fuzzifier and defuzzifier are designed for realizing this conversion. The fuzzifer is defined as a mapping from a real-valued point x^* to a fuzzy set A', while the defuzzifier converts fuzzy set B' to crisp point y^*.

III. FUNCTIONAL EQUIVALENCE BETWEEN RBF NN & FS

Under some assumptions in the fuzzy inference process, the functional equivalence between RBF NN and FS can be built up in the feed-forward calculation of the neural network.

Assume:

1) For the fuzzy rule depicted by (3), all the membership functions in the premise are Gaussian functions with the same variance, while the center and the height of the fuzzy set $\mu_{w_i}(y)$ are w_i and 1 respectively.
2) t-norm operator is multiplication.
3) The fuzzifier uses singleton method, that is, if real world crisp input is $x^*=[x_1^*, x_2^*, \ldots, x_n^*]$, then:

$$\mu_{A'}(x)=\begin{cases}1 & x=x^*\\ 0 & x\neq x^*\end{cases} \tag{6}$$

4) The defuzzifier is defined as:

$$y^*=\sum_{i=1}^{M} height(\mu_{B'}(y))\cdot center(\mu_{B'}(y)) \tag{7}$$

Therefore the fuzzy inference process can be rewritten as:

(1) The membership function of the premise:

$$\mu(x)=t\left[\mu_{c_{i1}}(x_1),\mu_{c_{i2}}(x_2),\ldots,\mu_{c_{in}}(x_n)\right]$$
$$=e^{-\frac{(x_1-c_{i1})^2}{\sigma_i^2}}\cdot e^{-\frac{(x_2-c_{i2})^2}{\sigma_i^2}}\cdot\ldots\cdot e^{-\frac{(x_n-c_{in})^2}{\sigma_i^2}}=\exp\left(-\frac{\left\|x-c_i\right\|^2}{\sigma_i^2}\right)$$

(2) The membership function of the IF-THEN fuzzy rule

$$\mu_{x\to y}(x,y)=\mu(x)\cdot\mu_{w_i}(y)=\exp\left(-\frac{\left\|x-c_i\right\|^2}{\sigma_i^2}\right)\cdot\mu_{w_i}(y)$$

(3) Since the fuzzifier is singleton, the fuzzy set output B' is [12]:

$$\mu_{B'}(y)=\sup_x\left\{t\left[\mu_{A'}(x),\mu_{x\to y}(x,y)\right]\right\}$$
$$=\mu(x^*)\cdot\mu_{w_i}(y)=\exp\left(-\frac{\left\|x^*-c_i\right\|^2}{\sigma_i^2}\right)\cdot\mu_{w_i}(y)$$

(4) If the number of the rules is equal to the number of the hidden neurons, the crisp output of the defuzzifier is:

$$y^*=\sum_{i=1}^{M}\exp\left(-\frac{\left\|x^*-c_i\right\|^2}{\sigma_i^2}\right)\cdot w_i=\sum_{i=1}^{n_h}w_i\cdot\varphi_i(x^*)$$

It can be seen that the functional equivalence between RBF NN and FS in the feed-forward calculation is established. As compared with the derivation process in [6], some improvements have been made in our study. The normal fuzzy rule is used to express the knowledge base, while in [6] the conclusion part of the fuzzy rule is limited to be a constant, which is only a special case of the normal fuzzy proposition. Moreover [6] only calculates the membership function of the premise of the fuzzy rule as the weight of the corresponding rule and gets the overall output of the FS by weighted sum of each rule's output. This inference process ignores the implication of the fuzzy rule and thus is not so strict. In this paper, the most widely used implication method of the fuzzy rule is adopted. In addition, both inputs and outputs of the RBF NN are crisp values in real world. The functional equivalence implies that inputs and outputs of the FS should be crisp values too. Therefore the fuzzifier and the defuzzifier should be considered in the FS as we do in our study. It is clear that the derivation process in this paper is normal, strict and complete.

For FSE problem, the functional equivalence in feed-forward calculation is of significant meanings. This enables us to convert the RBF NN into explicit fuzzy rules and vice versa, which makes the working process of RBF NN transparent and understandable to the operators in the control center. Besides we can check whether the neural network includes all the typical fault scenarios through examining whether the corresponding FS is complete. If there is any rare fatal fault scenario omitted, the RBF NN can be mended with priori domain knowledge. This is helpful for improving the diagnosis capability of RBF NN.

It should be pointed out that the characteristics of the FSE makes the functional equivalence hold not only in the feed-forward calculation, but also throughout the training process of RBF NN. As shown by Andersen, etc [7], although there exists functional equivalence between RBF NN and FS, the RBF NN cannot be converted back to the corresponding FS after it is retrained, unless the following two conditions are satisfied.

(1) All the membership functions in the premises of different fuzzy rules must have the same variance, i.e. $\sigma_1=\sigma_2=\ldots=\sigma_i=\ldots=\sigma_{n_h}$;

(2) Rules which share a membership function, or centers with a same element, must be aligned in the input-dimension on which that function is defined throughout the training process of the RBF network (Fig. 3).

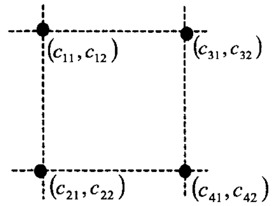

Fig. 3 The illustration of the alignment
$c_{11}=c_{21}, c_{31}=c_{41}$
$c_{12}=c_{32}, c_{22}=c_{42}$

These two conditions can be easily satisfied by the characteristics of the FSE. As pinpointed in section II.A, the spreads of all RBF neurons are same for FSE [1] and this is consistent with condition 1. Likewise condition 2 is guaranteed since the centers of the RBF neurons are binary variables, which keeps the alignment of the centers naturally. Therefore, for FSE problem, no matter whenever the knowledge is extracted from the RBF NN, it can be inserted back to this RBF NN even though it is retrained. Based on this point, a novel retraining strategy for RBF NN is presented in the case of power network topology change or power network expansion, which is described in detail below.

IV. RETRAINING STRATEGY FOR RBF NN

The topology structure of the power network may change due to restoration, maintenance and etc. At the same time, power network expansion may occur with the power system development. The trained RBF NN is expected to adapt these two cases instead of being totally retrained.

In the case of the power network topology change, the coordination relationship of the protective devices is affected, and thus some training samples, which were correct for the original power network structure, are not correct any more and their effects on the trained RBF NN should be eliminated. Besides some new training samples about the changing part of the power network need to be learned by the RBF NN. For power network expansion, since there are some new elements in the power network, some new inputs and outputs of the RBF NN have to be considered too.

Aiming at these two cases, the retraining strategy is proposed based on the functional equivalence between RBF NN and FS throughout the neural network training process. A simple power network (Fig. 4) is used as the illustrative example to explain the idea step by step. Suppose the corresponding RBF NN has been trained by typical fault scenarios.

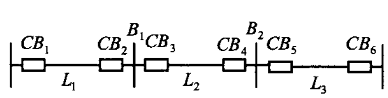

Fig. 4 The illustrative power network

Case 1: power network topology change. Assume L_3 is in maintenance and stop running (Fig. 5).

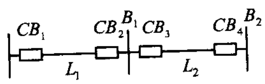

Fig. 5 The illustrative power network with topology change

Step 1: through knowledge extraction, acquire and store the knowledge, which is not related with the topology change, from the trained RBF NN.

Since one hidden neuron in the trained RBF NN is equivalent with one explicit fuzzy rule, the knowledge stored in the RBF NN can be extracted piece by piece according to the need. If L_3 is stop running, the knowledge about B_2, L_2 and L_3 is affected, while that of B_1 and L_1 is still correct. In this case, we use the centers of the hidden neurons as the inputs of the trained RBF NN and find the corresponding hidden neurons, which make the outputs of B_1 and L_1 equal 1. 7 hidden neurons imply the knowledge about B_1 and L_1 and can be stored as explicit fuzzy rules.

Step 2: learn the new knowledge about the changing part of the power network.

Since L_3 is stop running, the new training samples of B_2 and L_2 are constructed and used to train a new RBF NN.

Step 3: through knowledge insertion, mend the new RBF NN with the priori knowledge obtained in step 1 to get the complete RBF NN for the topology-changed power network.

The knowledge about B_1 and L_1 is inserted back to the new RBF NN, i.e., 7 hidden neurons are added and the corresponding centers are obtained in step 1. With the knowledge of B_1 and L_1, the advanced RBF NN is complete for performing FSE of the power network in Fig. 5.

Case 2: power network expansion. Suppose a new transmission line L_4 is added to the power network (Fig. 6).

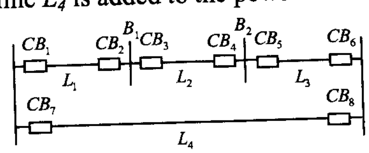

Fig. 6 The illustrative power network with network expansion

Once L_4 is added to the network, the coordination relationships of L_1, L_3 and L_4 are affected, while other elements, B_1, B_2 and L_2, remain unchanged. The operations of knowledge extraction and insertion are same with those of network topology change, however in this case some new inputs, CB_7, CB_8 and the protective relays of L_4, and the new output, L_4, should be considered in the process. Consequently the following step is added after step 1 and other steps, which remain unchanged, are omitted here.

Added step: the new inputs and outputs are added to the extracted knowledge in step 1.

Since the extracted knowledge of B_1, B_2 and L_2 are not related with the new element L_4, the corresponding weights of the new added inputs and outputs are all zero. With this operation, the extracted knowledge is extended and ready for insertion.

It can be seen that the suggested retraining strategy only retrain the neural network for the elements whose training samples are affected by the topology change or network expansion. Because of the local nature of the FSE, this part of the elements should be a small percentage of the whole power network and the larger the power network, the more obvious this effect. Consequently the retraining strategy can improve the adaptability and the efficiency of the RBF NN greatly.

V. COMPUTER SIMULATION RESULTS

A simple 4-bus power system is used as the test system (Fig. 7), in which there are 9 protected components: 4 buses, 1 transformer and 4 transmission lines. The protection relay system considered in the computer test is a simplified system, which includes transmission lines main protection (*MLP*) and backup protection (*BLP*), main protection for buses (*MBP*) and the transformer (*MTP*).

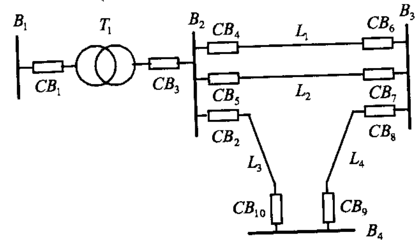

Fig. 7 The test power system

For the power network shown in Fig. 7, forty typical fault scenarios are worked out to constitute the training sample set. For each fault scenario, the states of all relays and circuit breakers are taken as the neural network inputs. The states of the 9 system components (4 buses, 1 transformer and 4 lines) are the outputs. If a certain output approaches to 1, then the corresponding component is considered in fault. After training, the trained RBF network has 33 input neurons, 37 hidden neurons and 9 output neurons, which can perform FSE with generalization capability [1].

Case 1: power network topology change, i.e. consider L_2 is in maintenance and stop running (Fig. 8).

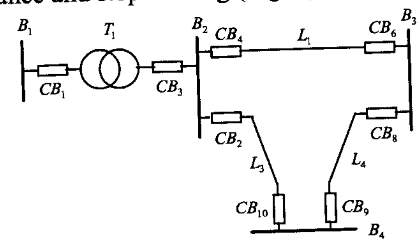

Fig. 8 The test power system with topology change

The training samples of L_1, L_2, B_2, B_3, L_3 and L_4 are affected and have to be reconstructed, while B_1, B_4 and T_1's remain unchanged. Therefore the knowledge (10 fuzzy rules) about B_1, B_4 and T_1 are extracted from the trained network. After training the new RBF NN by the reconstructed training samples, these 10 fuzzy rules are inserted and the complete RBF NN is obtained. The comparison results between the advanced and the general RBF NN are shown in Table 1.

Table 1 Comparison results between advanced and general RBF NN

	Advanced RBF NN	General RBF NN
mse of the training samples	1.9788×10^{-4}	8.0409×10^{-29}
Training time (s)	0.993	1.39
Number of the hidden neurons	30	31

Since the mean squared error (mse) between the calculated outputs and the desired outputs of the training samples is 1.9788×10^{-4}, the advanced RBF NN can realize accurate diagnosis for the training samples. On the premise of the accurate diagnosis, the training time of the advanced RBF NN is only 71.4% of that of the general RBF network and the larger the power network, the smaller the percentage. Table 1 demonstrates that the advanced RBF NN works successfully and efficiently for the case of power network topology change.

In order to examine the generalization capability of the designed advanced RBF NN we select other fault scenarios not existing in the training sample set for testing. Only 8 cases are listed in Table 2. All of them are severe cases with up to 2 mal-functions of relays and circuit breakers. The corresponding diagnosis results are listed in Table 3, in which each row is the output vector for one fault scenario. In our study, if a certain output in the output vector exceeds 0.5, then the corresponding component is considered in fault. From the output vectors, we can conclude that for all the test cases the faulted elements are recognized correctly.

Table 2 Severe test fault scenarios for topology change

Operated relays and tripped CBs	Fault element
1. MBP_1, BLP_3, CB_1	B_1
2. MBP_2, BLP_6, BLP_{10}, CB_3, CB_6, CB_{10}	B_2
3. MBP_3, MLP_8, CB_6, CB_8	B_3
4. MBP_4, BLP_2, CB_9, CB_{10}	B_4
5. MTP_1, BLP_{10}, CB_1, CB_3, CB_{10}	T_1
6. MLP_4, MLP_6, BLP_3, CB_4, CB_6	L_1
7. MLP_8, MLP_9, BLP_2, CB_8, CB_9	L_3
8. MLP_2, BLP_8, CB_2, CB_8	L_4

Case 2: power network expansion, i.e. consider a new generator-transformer block (B_5-T_2) is added (Fig. 9).

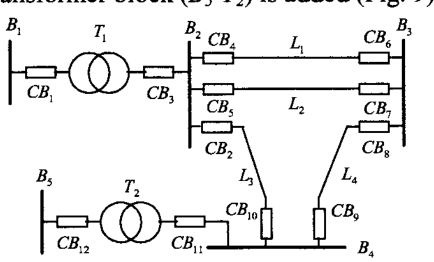

Fig. 9 The test power system with network expansison

Likewise, the knowledge (28 rules) about B_1, B_2, T_1, L_1 and L_2 are extracted and six new inputs (MTP_2, MBP_5, BLP_{11}, BLP_{12}, CB_{11}, CB_{12}) and two outputs (B_5, T_2) are added due to the network expansion. After the new RBF NN is trained to learn the knowledge about B_4, B_5, T_2, L_3 and L_4, the extracted knowledge is inserted back to complement the complete RBF NN. Similar results are obtained (Table 4). It can be seen that the advanced RBF NN cannot only diagnose the fault elements accurately but also improve the calculation efficiency greatly.

In order to test the generalization capability for this case, 15 severe fault scenarios with up to 2 mal-functions of relays and circuit breakers or up to 2 simultaneous faults are listed in Table 5. Table 6 gives the corresponding diagnosis results, which demonstrates that the advanced RBF NN has the generalization capability and is suitable for on-line FSE.

Table 4 Comparison results between advanced and general RBF NN

	Advanced RBF NN	General RBF NN
mse of the training samples	2.0725×10^{-28}	2.9188×10^{-28}
Training time (s)	1.064	2.065
Number of the hidden neurons	47	47

Table 5 Severe test fault scenarios for network expansion

Operated relays and tripped CBs	Fault element
1. MBP_1, BLP_3, CB_1	B_1
2. MBP_2, BLP_6, BLP_{10}, CB_3, CB_5, CB_6, CB_{10}	B_2
3. MBP_3, MLP_8, CB_6, CB_7, CB_8	B_3
4. MBP_4, BLP_2, CB_9, CB_{10}, CB_{11}	B_4
5. MTP_1, BLP_{10}, CB_1, CB_3, CB_{10}	T_1
6. MLP_4, MLP_6, BLP_3, CB_4, CB_6	L_1
7. MLP_7, BLP_1, BLP_6, BLP_{10}, CB_1, CB_6, CB_7, CB_{10}	L_2
8. MLP_8, MLP_9, BLP_{12}, CB_8, CB_9	L_3
9. MLP_2, MLP_{10}, BLP_{11}, CB_2, CB_{10}	L_4
10. MBP_5, BLP_{11}, CB_{12}	B_5
11. MTP_2, BLP_8, CB_{11}, CB_{12}	T_2
12. MTP_1, MLP_8, MLP_9, CB_1, CB_3, CB_8, CB_9	T_1, L_3
13. MBP_4, MLP_2, MLP_{10}, CB_2, CB_9, CB_{10}, CB_{11}	B_4, L_4
14. MLP_4, MLP_6, MLP_8, MLP_9, CB_4, CB_6, CB_8, CB_9	L_1, L_3
15. MTP_2, MLP_5, MLP_7, CB_5, CB_7, CB_{11}, CB_{12}	T_2, L_2

Table 3 The diagnosis results of the severe test fault scenarios for topology change

	B_1	B_2	B_3	B_4	T_1	L_1	L_2	L_3	L_4
1	0.9071	-0.0007	-0.0011	0.0034	0.0849	0.0025	0	0.0105	-0.0067
2	0.0326	1.2638	-0.0280	0.0140	-0.0290	-0.1241	0	0.0867	-0.2161
3	0.0335	0.0254	0.7394	-0.1807	-0.0167	-0.0302	0	0.4163	0.0129
4	-0.0595	-0.2347	-0.1283	1.2218	0.0627	0.0361	0	0.2427	-0.1408
5	-0.3093	0.3791	-0.1520	0.0306	0.9097	0.1871	0	0.0694	-0.1147
6	0.1825	0.0120	-0.0842	-0.0546	-0.0651	1.0116	0	0.1033	-0.1055
7	-0.0580	-0.2385	-0.0735	0.2576	0.0627	0.0139	0	1.1603	-0.1245
8	0.0471	0.1144	0.0717	0.2975	-0.0484	0.0060	0	-0.0910	0.6029

Table 6 The diagnosis results of the severe test fault scenarios for network expansion

	B_1	B_2	B_3	B_4	B_5	T_1	T_2	L_1	L_2	L_3	L_4
1	0.8742	-0.0050	0.0106	0.0059	0.0245	0.0991	-0.0141	0.0019	0.0019	-0.0003	0.0015
2	0.1045	1.0616	0.0514	0.0340	-0.1227	-0.0453	0.0260	-0.1055	0.0702	0.0033	-0.0774
3	0.0754	0.0352	0.8365	-0.1557	-0.1783	-0.0258	0.0206	-0.0188	-0.0188	0.4229	0.0069
4	0.0627	0.0111	0.0150	1.0158	-0.1058	-0.0184	0.0481	-0.0051	-0.0051	-0.0097	-0.0088
5	-0.1172	0.2663	-0.1497	0.0711	-0.1244	0.8450	0.0321	0.1033	0.1033	0.0309	-0.0608
6	0.3719	-0.0244	-0.0090	0.0108	-0.2534	-0.1434	0.0482	0.9749	0.0023	0.0108	0.0112
7	-0.0755	0.3082	0.0811	-0.0078	0.0509	0.0570	0.0077	-0.0575	0.6547	0.0042	-0.0230
8	0.1464	-0.1899	0.0249	0.3134	-0.1356	-0.0154	-0.1035	-0.0002	-0.0002	1.0387	-0.0786
9	-0.0262	-0.0097	-0.0137	0.0016	0.1583	0.0091	-0.1138	-0.0004	-0.0004	0.0128	0.9826
10	0.0523	0.0051	0.0077	0.0002	0.8552	-0.0180	0.0963	-0.0007	-0.0007	0.0020	0.0007
11	0.1302	0.0155	-0.1899	0.3735	-0.1001	-0.0584	0.8795	-0.0152	-0.0152	-0.0893	0.0693
12	-0.2717	-0.0927	-0.0956	-0.0174	-0.0379	0.9719	0.1733	0.0322	0.0322	0.9536	0.0519
13	0.1094	-0.0465	0.0433	0.6438	-0.4742	-0.0244	-0.1531	-0.0006	-0.0006	-0.0305	0.9335
14	-0.2610	-0.0833	-0.1805	0.0039	-0.0483	0.0896	0.1630	0.8541	0.1031	0.9220	0.0373
15	-0.2716	-0.0665	-0.0932	-0.0231	-0.3296	0.0900	1.0599	0.0050	0.9251	0.0635	0.0505

VI. CONCLUSIONS

In this paper, in order to enhance the adaptation capability of the RBF NN in the case of power network topology change or power network expansion, the functional equivalence between RBF NN and FS is established throughout the neural network training process. This enables us to construct the hybrid FSE system based on RBF NN and FS, which takes the advantages of both intelligent systems and provides a powerful tool for FSE problem. Furthermore, based on this equivalence, a novel retraining strategy is presented for RBF NN, which can extract the unchanged knowledge from the original RBF NN and then insert the knowledge back to the new RBF NN about the changing part of the power network in the case of network topology change or expansion. The retraining strategy has been implemented and tested in a 4-bus power system. The simulation results show that the advanced RBF NN works successfully and efficiently for FSE in power networks.

It should be pointed out that, based on our recent research, this approach as working with network partitioning technique can solve large-scale power network FSE problem fairly well. Corresponding results are included in a companion paper [2].

VII. ACKNOLEDGEMENTS

The research is supported by National Key Basic Research Special Fund (No. G1998020307), P.R. China and the University of Hong Kong, to whom sincere acknowledgements are expressed.

VIII. REFERENCES

[1] T.S. Bi, Z. Yan, F.S. Wen, Y.X. Ni, C. M. Shen, F. F. Wu and Q. X. Yang, "On-line fault section estimation in power systems with radial basis function neural network", accepted by International Journal of Electrical Power & Energy Systems.

[2] T. S. Bi, Y. X. Ni, C. M. Shen and F. F. Wu, "An on-line distributed intelligent fault section estimation system for large scale power networks", submitted to IEEE Transactions on Power Systems.

[3] J. Park and I.W. Sandberg, "Universal approximation using radial-basis-function networks", Neural Computation, vol. 3, 1991, pp. 246-257.

[4] Y. S. Abu-Mostafa, "Learning from hints in neural networks", Journal of Complexity, vol. 6, 1990, pp. 192-198.

[5] Volker Tresp, Jürgen Hollatz and Subutai Ahmad, "Representing probabilistic rules with networks of Gaussian basis functions", Machine Learning, vol. 27, 1997, pp. 173-200.

[6] J.S. Roger Jang and C.T. Sun, "Functional equivalence between radial basis function networks and fuzzy inference systems", IEEE Transactions on Neural Networks, vol. 4, no. 1, January 1993, pp. 156-159.

[7] H. C. Andersen, A. Lotfi and L.C. Westphal, "Comments on 'Functional equivalence between radial basis function networks and fuzzy inference systems'", IEEE Transactions on Neural Networks, vol. 9, no. 6, November 1998, pp. 1529-1531.

[8] J. Moody and C. Darken, "Fast learning in networks of locally-tuned processing units", Neural Computation, vol. 1, 1989, pp. 281-294.

[9] Simon Haykin, "Neural networks, a comprehensive foundation", New York, Prentice Hall, c1999, pp. 256-318.

[10] S. Lee and R. M. Kil, "A Gaussian potential function network with hierarchically self-organizing learning", Neural Networks, vol. 4, 1991, pp. 207-224.

[11] S. Chen, C.F.N. Cowan and P.M. Grant, "Orthogonal least squares learning algorithm for radial basis function networks", IEEE Transactions on Neural Networks, vol. 2, no. 2, March 1991, pp. 302-309.

[12] L.X. Wang, "A course in fuzzy systems and control", Prentice-Hall, 1997, pp. 1-118.

IX. BIOGRAPHIES

Tianshu Bi received her B. Eng, and M. Eng degrees from North China Electric Power University, P. R. China, in 1994 and 1997 respectively. She is currently a Ph. D. candidature at the Dept. of E.E.E., the University of Hong Kong. Her research area includes power system operation, protective relay and the application of artificial intelligence in power systems.

Yixin Ni received her B. Eng., M. Eng., and Dr. Eng. from Electrical Engineering Department, Tsinghua University, P. R. China in 1968, 1981 and 1983 respectively. Her research interest is in power system modeling, simulation, stability and control, and power electronics applications in power systems. She was a professor of Tsinghua University and is now with the University of Hong Kong.

Felix F. Wu joined the University of Hong Kong as Chair professor of Electrical Engineering in September 1995. Prior to that, he was Professor and Vice Chairman, Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, the same institute where he received his Ph.D. degree. His research interests are in power system planning and operation, including economics and reliability in system planning, real-time security assessment, and design of energy management systems and distribution automation. Recently he has been involved in the design of industry restructuring and electricity pricing.